INTERNATIONAL STANDARD

ISO/IEC 30112 WD12

ISO/IEC 30112 WD12
2018-02-12

Information technology — Specification methods for cultural conventions

Technologies de l'information —
Méthodes de modélisation des conventions culturelles

Contents **Page**

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

This International Standard was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee 35, *User Interfaces*.

Introduction

This International Standard defines general mechanisms to specify cultural conventions, and it defines formats for a number of specific cultural conventions in the areas of character classification and conversion, sorting, number formatting, monetary formatting, date formatting, message display, addressing of persons, postal address formatting, and telephone number handling.

There are a number of benefits coming from this International Standard:

Rigid specification	Using this International Standard, a user can rigidly specify a number of the cultural conventions that apply to the information technology environment of the user.
Cultural adaptability	If an application has been designed and built in a culturally neutral manner, the application may use the specifications as data to its APIs, and thus the same application may accommodate different users in a culturally acceptable way to each of the users, without change of the binary application.
Productivity	This International Standard specifies cultural conventions and how to specify data for them. With that data an application developer is relieved from getting the different information to support all the cultural environments for the expected customers of the product. The application developer is thus ensured of culturally correct behaviour as specified by the customer, and possibly more markets may be reached as customers may have the possibility to provide the data themselves for markets that were not targeted.
Uniform behaviour	When a number of applications share one cultural specification, which may be supplied from the user or provided by the application or operating system, their behaviour for cultural adaptation becomes uniform.

The specification formats are independent of platforms and specific encoding, and targeted to be usable from a wide range of programming languages.

A number of cultural conventions, such as spelling, hyphenation rules and terminology, are not specifiable with this International Standard, but the International Standard provides mechanisms to define new categories and also new keywords within existing categories. An internationalized application may take advantage of information provided with the FDCC-set (such as the language) to provide further internationalized services to the user.

This International Standard defines a format compatible with the one used in the International string ordering standard, ISO/IEC 14651. This International Standard is upward compatible with parts of the ISO/IEC 9945 POSIX standard,

especially those on POSIX locales and charmaps. The major extensions from that text are listed in annex A. This International Standard has enhanced functionality in a number of areas such as ISO/IEC 10646 support, more classification of characters, transliteration, dual (multi) currency support, enhanced date and time formatting, personal name writing, postal address formatting, telephone number handling, keyboard handling, and management of categories. There is enhanced support for character sets including ISO/IEC 2022 handling and an enhanced method to separate the specification of cultural conventions from an actual encoding via a description of the character repertoire employed. A standard set of values for all the categories has been defined covering the repertoire of ISO/IEC 10646.

Information technology — Specification methods for cultural conventions

1 SCOPE

This International Standard specifies description formats and functionality for the specification of cultural conventions, description formats for character sets, and description formats for binding character names to ISO/IEC 10646, plus a set of default values for some of these items.

The specification is upward compatible with POSIX locale specifications - a locale conformant to POSIX specifications will also be conformant to specifications in this International Standard, while the reverse condition will not hold. Some of the descriptions are intended to be coded in text files to be used via Application Programming Interfaces, that are expected to be developed for a number of systems which comply with ISO/IEC 9945. An alignment effort has been undertaken for this specification to be aligned with ISO/IEC 9945.

2 NORMATIVE REFERENCES

The following referenced documents are indispensible for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 646, *Information technology - ISO 7-bit coded character set for information interchange*.

ISO/IEC 14977, *Information technology - Syntactic metalanguage - Extended BNF*.

ISO 639 (all parts), *Codes for the representation of names of languages*.

ISO/IEC 2022, *Information technology - Character code structure and extension techniques*.

ISO 3166 (all parts), *Codes for the representation of names of countries and their subdivisions*.

ISO 15924, *Information and documentation -- Codes for the representation of names of scripts.*

ISO 4217, *Codes for the representation of currencies and funds*.

ISO 8601, *Data elements and interchange formats - Information interchange - Representation of dates and times*.

ISO/IEC 9899, *Information technology - Programming language C*.

ISO/IEC 9945, *Information technology - Portable Operating System Interface (POSIX)*.

ISO/IEC 9945-2:1993, *Information technology - Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities*.

ISO/IEC 10646, *Information technology - Universal Coded Character Set (UCS)*.

ISO/IEC 14651, *Information technology - International string ordering and comparison - Method for comparing character strings and description of the common template tailorable ordering*.

ISO/IEC 15897, *Information technology - Procedures for registration of cultural elements*.

3 TERMS, DEFINITIONS AND NOTATIONS

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1 Bytes and characters

3.1.1.1
byte
An individually addressable unit of data storage that is equal to or larger than an octet, used to store a character or a portion of a character.

Note: A byte is composed of a contiguous sequence of bits, the number of which is implementation defined. The least significant bit is called the low-order bit; the most significant bit is called the high-order bit

3.1.1.2
character
A member of a set of elements used for the organization, control or representation of data

3.1.1.3
coded character
A sequence of one or more bytes representing a single character

3.1.1.4
text file
A file that contains characters organized into one or more lines

3.1.2 cultural and other major concepts

3.1.2.1
cultural convention
A data item for information technology that may vary dependent on language, territory, or other cultural habits

3.1.2.2
FDCC
A Formal Definition of a Cultural Convention, that is a cultural convention put into a formal definition scheme

3.1.2.3
FDCC-set
A Set of Formal Definitions of Cultural Conventions (FDCC's). The definition of the subset of a user's information technology environment that depends on language and cultural conventions

Note: the FDCC-set is a superset of the "locale" term in C and POSIX.

3.1.2.4
charmap
A definition of a mapping between symbolic character names and character codes, plus related information

3.1.2.5
repertoiremap
A definition of a mapping between symbolic character names and characters for the repertoire of characters used in a FDCC-set

NOTE: This further described in clause 6.

3.1.3 FDCC categories related

3.1.3.1
character class:
A named set of characters sharing an attribute associated with the name of the class.

3.1.3.2
collation
The logical ordering of strings according to defined precedence rules

3.1.3.3
collating element
The smallest entity used to determine logical ordering

Note: See collating sequence. A collating element consists of either a single character, or two or more characters collating as a single entity. The LC_COLLATE category in the associated FDCC-set determines the set of collating elements.

3.1.3.4
multicharacter collating element
A sequence of two or more characters that collate as an entity

Note: For example, in some languages two characters are sorted as one letter, as in the case for Danish and Norwegian "aa".

3.1.3.5
collating sequence
The relative order of collating elements as determined by the setting of the LC_COLLATE category in the applied FDCC-set

3.1.3.6
equivalence class
A set of collating elements with the same primary collation weight

NOTE: Elements in an equivalence class are typically elements that naturally group together, such as all accented letters based on the same letter.

The collation order of elements within an equivalence class is determined by the weights assigned on any subsequent levels after the primary weight.

3.2 Notations

The following notations and common conventions for specifications apply to this International Standard:

3.2.1 Notation for defining syntax

In this International Standard, the description of an individual record in a FDCC-set is done using the syntax notation given in the following.

The syntax notation looks as follows:

"<format>",[<arg1>,<arg2>,...,<argn>]

The <format> is given in a format string enclosed in double quotes, followed by a number of parameters, separated by commas. It is similar to the format specification defined in the ISO/IEC 9945 standard and the format specification used in C language printf() function. The format of each parameter is given by an escape sequence as follows:

%s	specifies a string
%d	specifies a decimal integer
%c	specifies a character
%o	specifies an octal integer
%x	specifies a hexadecimal integer

A " " (an empty character position) in the syntax string represents one or more <blank> characters.

All other characters in the format string represent themselves, except:

%%	specifies a single %
\n	specifies an end-of-line

The notation "..." is used to specify that repetition of the previous specification is optional, and this is done in both the format string and in the parameter list.

3.2.3 Portable character set

A set of symbolic names for characters in Table 1, which is called the portable character set, is used in character description text of this specification. The first eight entries in Table 1 are defined in ISO/IEC 6429 and the rest is defined in ISO/IEC 9945 with some definitions from ISO/IEC 10646.

Table 1: Portable character set

Symbolic name	Glyph	UCS	Description

<NUL>		<U0000>	NULL (NUL)
<alert>		<U0007>	BELL (BEL)
<backspace>		<U0008>	BACKSPACE (BS)
<tab>		<U0009>	CHARACTER TABULATION (HT)
<carriage-return>		<U000D>	CARRIAGE RETURN (CR)
<newline>		<U000A>	LINE FEED (LF)
<vertical-tab>		<U000B>	LINE TABULATION (VT)
<form-feed>		<U000C>	FORM FEED (FF)
<space>		<U0020>	SPACE
<exclamation-mark>	!	<U0021>	EXCLAMATION MARK
<quotation-mark>	"	<U0022>	QUOTATION MARK
<number-sign>	#	<U0023>	NUMBER SIGN
<dollar-sign>	$	<U0024>	DOLLAR SIGN
<percent-sign>	%	<U0025>	PERCENT SIGN
<ampersand>	&	<U0026>	AMPERSAND
<apostrophe>	'	<U0027>	APOSTROPHE
<left-parenthesis>	(	<U0028>	LEFT PARENTHESIS
<right-parenthesis>	)	<U0029>	RIGHT PARENTHESIS
<asterisk>	*	<U002A>	ASTERISK
<plus-sign>	+	<U002B>	PLUS SIGN
<comma>	,	<U002C>	COMMA
<hyphen-minus>	-	<U002D>	HYPHEN-MINUS
<hyphen>	-	<U002D>	HYPHEN-MINUS
<full-stop>	.	<U002E>	FULL STOP
<period>	.	<U002E>	FULL STOP
<slash>	/	<U002F>	SOLIDUS
<solidus>	/	<U002F>	SOLIDUS
<zero>	0	<U0030>	DIGIT ZERO
<one>	1	<U0031>	DIGIT ONE
<two>	2	<U0032>	DIGIT TWO
<three>	3	<U0033>	DIGIT THREE
<four>	4	<U0034>	DIGIT FOUR
<five>	5	<U0035>	DIGIT FIVE
<six>	6	<U0036>	DIGIT SIX
<seven>	7	<U0037>	DIGIT SEVEN
<eight>	8	<U0038>	DIGIT EIGHT
<nine>	9	<U0039>	DIGIT NINE
<colon>	:	<U003A>	COLON
<semicolon>	;	<U003B>	SEMICOLON
<less-than-sign>	<	<U003C>	LESS-THAN SIGN
<equals-sign>	=	<U003D>	EQUALS SIGN
<greater-than-sign>	>	<U003E>	GREATER-THAN SIGN
<question-mark>	?	<U003F>	QUESTION MARK
<commercial-at>	@	<U0040>	COMMERCIAL AT
<A>	A	<U0041>	LATIN CAPITAL LETTER A
<B>	B	<U0042>	LATIN CAPITAL LETTER B
<C>	C	<U0043>	LATIN CAPITAL LETTER C
<D>	D	<U0044>	LATIN CAPITAL LETTER D
<E>	E	<U0045>	LATIN CAPITAL LETTER E
<F>	F	<U0046>	LATIN CAPITAL LETTER F
<G>	G	<U0047>	LATIN CAPITAL LETTER G
<H>	H	<U0048>	LATIN CAPITAL LETTER H
<I>	I	<U0049>	LATIN CAPITAL LETTER I
<J>	J	<U004A>	LATIN CAPITAL LETTER J
<K>	K	<U004B>	LATIN CAPITAL LETTER K
<L>	L	<U004C>	LATIN CAPITAL LETTER L
<M>	M	<U004D>	LATIN CAPITAL LETTER M
<N>	N	<U004E>	LATIN CAPITAL LETTER N
<O>	O	<U004F>	LATIN CAPITAL LETTER O

<P>	P	<U0050>	LATIN CAPITAL LETTER P
<Q>	Q	<U0051>	LATIN CAPITAL LETTER Q
<R>	R	<U0052>	LATIN CAPITAL LETTER R
<S>	S	<U0053>	LATIN CAPITAL LETTER S
<T>	T	<U0054>	LATIN CAPITAL LETTER T
<U>	U	<U0055>	LATIN CAPITAL LETTER U
<V>	V	<U0056>	LATIN CAPITAL LETTER V
<W>	W	<U0057>	LATIN CAPITAL LETTER W
<X>	X	<U0058>	LATIN CAPITAL LETTER X
<Y>	Y	<U0059>	LATIN CAPITAL LETTER Y
<Z>	Z	<U005A>	LATIN CAPITAL LETTER Z
<left-square-bracket>	[	<U005B>	LEFT SQUARE BRACKET
<backslash>	\	<U005C>	REVERSE SOLIDUS
<reverse-solidus>	\	<U005C>	REVERSE SOLIDUS
<right-square-bracket>	]	<U005D>	RIGHT SQUARE BRACKET
<circumflex-accent>	^	<U005E>	CIRCUMFLEX ACCENT
<circumflex>	^	<U005E>	CIRCUMFLEX ACCENT
<low-line>	_	<U005F>	LOW LINE
<underscore>	_	<U005F>	LOW LINE
<grave-accent>	`	<U0060>	GRAVE ACCENT
<a>	a	<U0061>	LATIN SMALL LETTER A
<b>	b	<U0062>	LATIN SMALL LETTER B
<c>	c	<U0063>	LATIN SMALL LETTER C
<d>	d	<U0064>	LATIN SMALL LETTER D
<e>	e	<U0065>	LATIN SMALL LETTER E
<f>	f	<U0066>	LATIN SMALL LETTER F
<g>	g	<U0067>	LATIN SMALL LETTER G
<h>	h	<U0068>	LATIN SMALL LETTER H
<i>	I	<U0069>	LATIN SMALL LETTER I
<j>	j	<U006A>	LATIN SMALL LETTER J
<k>	k	<U006B>	LATIN SMALL LETTER K
<l>	l	<U006C>	LATIN SMALL LETTER L
<m>	m	<U006D>	LATIN SMALL LETTER M
<n>	n	<U006E>	LATIN SMALL LETTER N
<o>	o	<U006F>	LATIN SMALL LETTER O
<p>	p	<U0070>	LATIN SMALL LETTER P
<q>	q	<U0071>	LATIN SMALL LETTER Q
<r>	r	<U0072>	LATIN SMALL LETTER R
<s>	s	<U0073>	LATIN SMALL LETTER S
<t>	t	<U0074>	LATIN SMALL LETTER T
<u>	u	<U0075>	LATIN SMALL LETTER U
<v>	v	<U0076>	LATIN SMALL LETTER V
<w>	w	<U0077>	LATIN SMALL LETTER W
<x>	x	<U0078>	LATIN SMALL LETTER X
<y>	y	<U0079>	LATIN SMALL LETTER Y
<z>	z	<U007A>	LATIN SMALL LETTER Z
<left-brace>	{	<U007B>	LEFT CURLY BRACKET
<left-curly-bracket>	{	<U007B>	LEFT CURLY BRACKET
<vertical-line>	|	<U007C>	VERTICAL LINE
<right-brace>	}	<U007D>	RIGHT CURLY BRACKET
<right-curly-bracket>	}	<U007D>	RIGHT CURLY BRACKET
<tilde>	~	<U007E>	TILDE

This International Standard may use other symbolic character names than the above in examples, to illustrate the use of the range of symbols allowed by the syntax specified in 4.1.1.

4 FDCC-set

A FDCC-set is the definition of the subset of a user's information technology environment that depends on language and cultural conventions. A FDCC-set is made up from one or more categories. Each category is identified by its name and controls specific aspects of the behaviour of components of the system. The functionality is implied by the description of the categories. This International Standard defines the following categories:

LC_IDENTIFICATION	Versions and status of categories
LC_CTYPE	Character classification, case conversion and code transformation.
LC_COLLATE	Collation order.
LC_TIME	Date and time formats.
LC_NUMERIC	Numeric, non-monetary formatting.
LC_MONETARY	Monetary formatting.
LC_MESSAGES	Formats of informative and diagnostic messages and interactive responses.
LC_XLITERATE	Character transliteration.
LC_NAME	Format of writing personal names.
LC_ADDRESS	Format of postal addresses.
LC_TELEPHONE	Format for telephone numbers, and other telephone information.
LC_PAPER	Paper format
LC_MEASUREMENT	Information on measurement system
LC_KEYBOARD	Format for identifying keyboards.

Note: In future editions of this International Standard further categories may be added.

Other category names beginning with the 3 characters "LC_" are reserved for future standardization, except for category names beginning with the five characters "LC_X_" which is not used for future addition of categories specified in this International Standard. An application may thus use category names beginning with the five characters "LC_X_" for application defined categories to avoid clashes with future standardized categories.

This International Standard also defines an FDCC-set named "i18n" with values for some of the above categories in order to simplify FDCC-set descriptions for a number of cultures. The contents of "i18n" categories should not necessarily be considered as the most commonly accepted values, while in many cases it could be the recommended values. The complete "i18n" FDCC-set is defined as the sum of the "i18n" categories specified in the clauses below. The ”i18n” FDCC-set and its parts are released under the GNU Public License, version 2, as it is taken from glibc sources.

4.1 FDCC-set description

FDCC-sets are described with the syntax presented in this subclause. For the purposes of this International Standard, the text is referred to as the FDCC-set definition text or FDCC-set source text.

The **FDCC-set definition text** contains one or more FDCC-set category source definitions, and does not contain more than one definition for the same FDCC-set category. If the text contains source definitions for more than one category, application-defined categories, if present, appears after the categories defined by this clause. A category source definition contains either the definition of a category or a copy directive. In the event that some of the information for a FDCC-set category, as specified in this International Standard, is missing from the FDCC-set source definition, the behaviour of that category, if it is referenced, is unspecified. A FDCC-set category is the normal way of specifying a single FDCC.

There are no **naming conventions** for FDCC-sets specified in this International Standard, but clause 6.8 in ISO/IEC 15897:1999 specifies naming rules for POSIX locales, charmaps and repertoiremaps, that may also be applied to FDCC-sets, charmaps and repertoiremaps specified according to this International Standard.

A **category source definition** consists of a category header, a category body, and a category trailer. A category header consists of the character string naming of the category, beginning with the characters "LC_". The category trailer consists of the string "END", followed by one or more "blank"s and the string used in the corresponding category header.

The **category body** consists of one or more lines of text. Each line is one of the following:

- a line containing an identifier, optionally followed by one or more operands. Identifiers are either keywords, identifying a particular FDCC, or collating elements, or section symbols,
- one of transliteration statements defined in 4.3.

In addition to the keywords defined in this International Standard, the source can contain application-defined keywords. Each **keyword** within a category has a unique name (i.e., two categories can have a commonly-named keyword); no keyword starts with the characters "LC_". Identifiers are separated from the operands by one or more "blank"s.

Operands are characters, collating elements, section symbols, or strings of characters. Strings are enclosed in double-quotes. Literal double-quotes within strings are preceded by the <escape character>, described below. When a keyword is followed by more than one operand, the operands are separated by semicolons; "blank"s are allowed before and/or after a semicolon.

4.1.1 Character representation

Individual characters, characters in strings, and collating elements are represented using symbolic names, UCS notation or characters themselves, or

as octal, hexadecimal, or decimal constants as defined below. When constant notation is used, the resultant FDCC-set definitions need not be portable between systems.

(0) The left angle bracket (<) is a reserved symbol, denoting the start of a symbolic name; when used to represent itself outside a symbolic name it is preceded by the escape character.

(1) A character can be represented via a **symbolic name**, enclosed within angle brackets (< and >). The symbolic name, including the angle brackets, exactly matches a symbolic name defined in a charmap or a repertoiremap to be used, and is replaced by a character value determined from the value associated with the symbolic name in the charmap or a value associated via a repertoiremap. Repertoiremaps have predefined symbolic names for UCS characters, see clause 6. A FDCC-set may also use the UCS notation of clause 6 to represent characters, without a repertoiremap being defined for the FDCC-set. Use of the escape character or a right angle bracket within a symbolic name is invalid unless the character is preceded by the escape character.

Example: <c>;<c-cedilla> "<M><a><y>"

The items (2), (3), (4) and (5) are deprecated and are retained for compatibility with the POSIX standard. FDCC-sets should be specified in a coded character set independent way, using symbolic names. To make actual use of the FDCC-set, it is used together with charmaps and/or repertoiremaps, so that the symbolic character names can be resolved into the actual character encoding used.

(2) A character can be represented by the character itself, in which case the value of the character is application-defined. Within a string, the double-quote character, the escape character, and the right angle bracket character are escaped (preceded by the escape character) to be interpreted as the character itself. Outside strings, the characters

, ; < > escape_char

are escaped by the escape character to be interpreted as the character itself.

Example: c ä "May"

(3) A character can be represented as an octal constant. An octal constant is specified as the escape character followed by two or more octal digits. Each constant represents a byte value.

Example: \143; \347; "\115"

(4) A character can be represented as a hexadecimal constant. A hexadecimal constant is specified as the escape character followed by an x followed by two or more hexadecimal digits. Each constant represents a byte value.

Example: \x63;\xe7;

(5) A character can be represented as a decimal constant. A decimal constant is specified as the escape character followed by a d followed by two or more decimal digits. Each constant represents a byte value.

Example: \d99; \d231;

(6) Multibyte characters can be represented by concatenated constants specified in byte order with the last constant specifying the least significant byte of the character. Concatenated constants can include a mix of the above character representations.

Example: \143\xe7; "\115\xe7\d171"

Only characters existing in the character set for which the FDCC-set definition is created are specified, whether using symbolic names, the characters themselves, or octal, decimal, or hexadecimal constants. If a charmap is present, only characters defined in the charmap can be specified using octal, decimal, or hexadecimal constants. Symbolic names not present in the charmap can be specified and are ignored, as specified under item (1) above.

Note: The <character> symbolic character notation is recommended for use of specifying all characters in a FDCC-set, to facilitate portability of the FDCC-sets, as the coded character set of the application of the FDCC-set may be different from the coded character set of the FDCC-set source. This is also recommended for format effectors in strings, such as in LC_DATE or LC_ADDRESS, where the format effectors are allowed to be stored together with the rest of the string, in a binary string with a different encoding from that of the source FDCC-set.

4.1.2 Continuation of lines

A line in a specification can be continued by placing an escape character as the last visible graphic character on the line; this continuation character is discarded from the input. The line is continued to the next non-comment line.

4.1.3 Names for copy keyword

In most of the categories a "copy" keyword is allowed. The name specified with this copy keyword is one of:

- "i18n" which indicate the "i18n" FDCC-set defined in this specification,
- the name of a FDCC-set or POSIX locale registered by the process defined in

ISO/IEC 15897,
- any other name which may be recognized in some local context - not being recommended as an international specification.

4.1.4 Pre-category statements

In a FDCC-set the following statements can precede category specifications, and they apply to all categories in the specified FDCC-set.

4.1.4.1 comment_char

The following line in a FDCC-set modifies the comment character. It has the following syntax, starting in column 1:

"comment_char %c\n", <comment_character>

The comment character defaults to the number-sign (#). All examples in this International Standard use "%" as the <comment_character>, except where otherwise noted. Blank lines and lines containing the <comment_character> in the first position are ignored. In collating statements a <comment_character> occurring where the delimiter ";" may occur, terminates the collating statement.

4.1.4.2 escape_char

The following line in a FDCC-set modifies the escape character to be used in the text. It has the following syntax, starting in column 1:

"escape_char %c\n", <escape_character>

The escape character is used for representing characters in 4.1.1 and for continuing lines.
The escape character defaults to backslash "\". All examples in this International Standard uses "/" as the escape character, except where otherwise noted.

4.1.4.3 repertoiremap

The following line in a FDCC-set specifies the name of a repertoiremap used to define the symbolic character names in the FDCC-set. There may be at most one "repertoiremap" line. It has the following syntax, starting in column 1:
"repertoiremap %s\n", <repertoiremap>

The name is one of:
- "i18nrep" which indicates the "i18nrep" repertoiremap defined in this specification,
- the name of a <repertoiremap> registered by the process defined in ISO/IEC

15897,
- any other name which may be recognized in some local context - not being recommended as an international specification.

4.1.4.4 charmap

The following line in a FDCC-set specifies the name of a charmap which may be used with the FDCC-set. It has the following syntax, starting in column 1:

"charmap %s\n",<charmap>

This keyword gives a hint on which charmaps a FDCC-set is meant to be supported by. There may be more than one charmap specification useful with a FDCC-set. It is an application's responsibility to decide what charmap specification is to be used with that application.

The name is one of:
- the name of a <charmap> registered by the process defined in ISO/IEC 15897,
- any other name which may be recognized in some local context - not being recommended as an international specification.

4.2 LC_IDENTIFICATION

The LC_IDENTIFICATION category defines properties of the FDCC-set, and which specification methods the FDCC-set is conforming to. Values must be supplied for all unless otherwise noted, and the operands are strings. The following keywords are defined:

title	Title of the FDCC-set.
source	Organization name of provider of the source.
address	Organization postal address.
contact	Name of contact person. This keyword is optional.
email	Electronic mail address of the organization, or contact person. This keyword is optional.
tel	Telephone number for the organization, in international format. This keyword is optional.
fax	Fax number for the organization, in international format. This keyword is optional.
language	Natural language to which the FDCC-set applies, as specified in ISO 639. If a two-letter code exists for this language, it is used, else the three-letter code is used. This keyword is optional.
territory	The geographic extent where the FDCC-set applies (where applicable), as two-letter form of ISO 3166. This keyword is optional.
script	Script that the FDCC-set especially uses, as defined by ISO/IEC 15924 and its registry. This keyword is optional.
audience	If not for general use, an indication of the intended user audience. This keyword is optional.
application	If for use of a special application, a description of the application. This keyword is optional.
abbreviation	Short name for provider of the source. This keyword is optional.
revision	Revision number consisting of digits and zero or more full stops (".") .
date	Revision date in the format according to this example: "1995-02-05" meaning the 5th of February, 1995.

If required information is not present in ISO 639 or ISO 3166, the string should be given as empty, and the relevant Maintenance Authority should be approached to get the needed item registered.

Note: Only one language per territory can be addressed with a single FDCC-set; an additional FDCC-set is required for each additional language for that territory.

category	Is used to define that a category is present and what specification the category is claiming conformance to. The first operand is a string in double-quotes that

describes the specification that the category is claiming conformance to, and the following values are defined:

”i18n:2004”
"i18n:2012"
“i18n:2018”
"posix:1993"

The second operand is a string with the category name, where the category names of clause 4 are defined. More than one "category" keyword may be given, but only one per category name.

The "i18n" LC_IDENTIFICATION category is:

```
LC_IDENTIFICATION
% This is the ISO/IEC TR 30112 "i18n" definition for
% the LC_IDENTIFICATION category.
%
title                   "ISO/IEC TR 30112 i18n FDCC-set"
source                  "ISO/IEC Copyright Office"
address                 "Case postale 56, CH-1211 Geneve 20, Switzerland"
contact                 ""
email                   ""
tel                     ""
fax                     ""
language                ""
territory               ""
revision                "1.1"
date                    "2010-07-30"
 %
 category  "i18n:2004";LC_IDENTIFICATION
 category  "i18n:2012";LC_CTYPE
 category  "i18n:2004";LC_COLLATE
 category  "i18n:2004";LC_TIME
 category  "i18n:2004";LC_NUMERIC
 category  "i18n:2004";LC_MONETARY
 category  "i18n:2004";LC_MESSAGES
 category  "i18n:2004";LC_NAME
 category  "i18n:2004";LC_ADDRESS
 category  "i18n:2004";LC_TELEPHONE
 category  "i18n:2012";LC_PAPER
 category  "i18n:2012";LC_MEASUREMENT
 category  "i18n:2012";LC_KEYBOARD

 END LC_IDENTIFICATION
```


4.3 LC_CTYPE

The LC_CTYPE category defines character classification, case conversion, character transformation, and other character attribute mappings. Support for the portable character set is required.

A series of characters in a specification can be represented by the hexadecimal symbolic ellipsis symbol ".." (two dots), the decimal symbolic ellipses symbols "...." (4 dots), the double increment hexadecimal symbolic ellipses "..(2)..", or the absolute ellipses "..." (3 dots).

The **hexadecimal symbolic ellipsis** ("..") specification is only valid between symbolic character names. The symbolic names consists of zero or more nonnumeric characters from the set shown with visible glyphs in Table 1 of clause 3.2.3, followed by an integer formed by one or more hexadecimal digits, using uppercase letters only for the range "A" to "F". The characters preceding the hexadecimal integer are identical in the two symbolic names, and the integer formed by the hexadecimal digits in the second symbolic name are identical to or greater than the integer formed by the hexadecimal digits in the first name. This is interpreted as a series of symbolic names formed from the common part and each of the integers in hexadecimal format using uppercase letters only between the first and the second integer, inclusive, and with a length of the symbolic names generated that is equal to the length of the first (and also the second) symbolic name. As an example, <U010E>..<U0111> is interpreted as the symbolic names <U010E>, <U010F>, <U0110>, and <U0111>, in that order.

The **decimal symbolic ellipsis** ("....") specification is only valid between symbolic character names. The symbolic names consist of zero or more nonnumeric characters from the set shown with visible glyphs in Table 1 of clause 3.2.3, followed by an integer formed by one or more decimal digits. The characters preceding the decimal integer are identical in the two symbolic names, and the integer formed by the decimal digits in the second symbolic name is identical to or greater than the integer formed by the decimal digits in the first name. This is interpreted as a series of symbolic names formed from the common part and each of the integers in decimal format between the first and the second integer, inclusive, and with a length of the symbolic names generated that is equal to the length of the first (and also the second) symbolic name. As an example, <j0101>....<j0104> is interpreted as the symbolic names <j0101>, <j0102>, <j0103>, and <j0104>, in that order.

The **double increment hexadecimal symbolic ellipses** ("..(2)..") works like the hexadecimal symbolic ellipses, but generates only every other of the symbolic character names. As an example. <U01AC>..(2)..<U01B2> is interpreted as the symbolic character names <U01AC>, <U01AE>, <U01B0>, and <U01B2>, in that order.

The **absolute ellipsis** specification is only valid within a single encoded character set. An ellipsis is interpreted as including in the list all characters

with an encoded value higher than the encoded value of the character preceding the ellipsis and lower than the encoded value of the character following the ellipsis. The absolute ellipsis specification is deprecated, as this is only relevant to FDCC-sets not using symbolic characters.
As an example, \x30;...;\x39 includes in the character class all characters with encoded values between the endpoints.

4.3.1 Character classification keywords

The following keywords are recognized. In the descriptions, the term "automatically included" means that it is not an error to either include the referenced characters or to omit them; the interpreting system provides them if missing and accept them silently if present.

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.
upper	Define characters to be classified as uppercase letters. No character specified for the keywords "cntrl", "digit", "punct", or "space" is specified. The uppercase letters A through Z of the portable character set, automatically belong to this class, with application-defined character values. The keyword may be omitted.
lower	Define characters to be classified as lowercase letters. No character specified for the keywords "cntrl", "digit", "punct", or "space" is specified. The lowercase letters a through z of the portable character set, automatically belong to this class, with application-defined character values. The keyword may be omitted.
alpha	Define characters to be classified as used to spell out the words for natural languages; such as letters, syllabic or ideographic characters. No character specified for the keywords "cntrl", "digit", "punct", or "space" is specified. In addition, characters classified as either "upper" or "lower" automatically belong to this class. The keyword may be omitted.
digit	Define the characters to be classified as decimal digits. Digits corresponding to the values 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 can be specified in groups of 10 digits, and in ascending order of the values they represent. The digits of the portable character set are automatically included. If this keyword is not specified, the digits 0 through 9 of the portable character set automatically belong to this class, with application-defined character values. The "digit" keyword is used to specify which characters are accepted as digits in input to an application, such as characters typed in or scanned in

from an input text file, and should list digits used with all the scripts supported by the FDCC-set. The keyword may be omitted.

alnum Define the characters to be classified as used to spell out the words for natural languages, and numeric digits. The characters of the "alpha" and "digit" classes are automatically included in this class. The keyword may be omitted.

outdigit Define the characters to be classified as decimal digits for output from an application, such as to a printer or a display or a output text file. Decimal digits corresponding to the values <0>, <1>, <2>, <3>, <4>, <5>, <6>, <7>, <8>, and <9> can be specified, and in ascending order of the values they represent. The intended use is for all places where decimal digits are used for output, including numeric and monetary formatting, and date and time formatting. Only one set of 10 decimal digits may be specified. If this keyword is not specified, the decimal digits 0 through 9 of the portable character set automatically belong to this class, with application-defined character values. The keyword may be omitted.

blank Define characters to be classified as "blank" characters. If this keyword is unspecified, the characters <space> and <tab>, with application-defined character values, belong to this character class.

space Define characters to be classified as white-space characters, to find syntactical boundaries. No character specified for the keywords "upper", "lower", "alpha", "digit", "graph", or "xdigit" is specified. If this keyword is not specified, the characters <space>, <form-feed>, <newline>, <carriage-return>, <tab>, and <vertical-tab>, automatically belong to this class, with application-defined character values. Any characters included in the class "blank" are automatically included. The class should not include the NO-BREAK spaces characters <U00A0>, <U2007>, <UFEFF>, as these characters should not be used for word boundaries. The keyword may be omitted.

cntrl Define characters to be classified as control characters. No character specified for the keywords "upper", "lower", "alpha", "digit", "punct", "graph", "print", or "xdigit" is specified. The keyword is specified.

punct Define characters to be classified as punctuation characters. No character specified for the keywords "upper", "lower", "alpha", "digit", "cntrl", "xdigit", or as the <space> character is specified. The keyword is specified.

xdigit	Define the characters to be classified as hexadecimal digits. Only the characters defined for the class "digit" are specified, in ascending sequence by numerical value, followed by sets of six characters representing the hexadecimal digits 10 through 15 in ascending order (for example <A>, <B>, <C>, <D>, <E>, <F>, <a>, <b>, <c>, <d>, <e>, <f>). The digits <0> through <9>, the uppercase letters <A> through <F>, and the lowercase letters <a> through <f>, automatically belong to this class, with application-defined character values.
graph	Define characters to be classified as printable characters, not including the <space> character. If this keyword is not specified, characters specified for the keywords "upper", "lower", "alpha", "digit", "xdigit", and "punct" belong to this character class. No character specified for the keyword "cntrl" is specified.
print	Define characters to be classified as printable characters, including the <space> character. If this keyword is not provided, characters specified for the keywords upper, lower, alpha, digit, xdigit, punct, graph, and the <space> character belong to this character class. No character specified for the keyword "cntrl" is specified.
toupper	Define the mapping of lowercase letters to uppercase letters. The operand consists of character pairs, separated by semicolons. The characters in each character pair are separated by a comma and the pair enclosed by parentheses. The first character in each pair is the lowercase letter, the second the corresponding uppercase letter. Only characters specified for the keywords "lower" and "upper" are specified. If this keyword is not specified, the lowercase letters <a> through <z>, and their corresponding uppercase letters <A> through <Z>, are automatically included, with application-defined character values.
tolower	Define the mapping of uppercase letters to lowercase letters. The operand consists of character pairs, separated by semicolons. The characters in each character pair are separated by a comma and the pair enclosed by parentheses. The first character in each pair is the uppercase letter, the second the corresponding lowercase letter. Only characters specified for the keywords "lower" and "upper" are specified. If this keyword is specified, the uppercase letters <A> through <Z>, and their corresponding lowercase letter, are specified. If this keyword is not specified, the mapping is the reverse mapping of the

one specified for toupper.

class Define characters to be classified in the class with the name given in the first operand, which is a string. This string only contains characters of the portable character set that either has the string "LETTER" in its description, or is a digit or <hyphen-minus> or <low-line>. The following operands are characters. This keyword is optional. The keyword can only be specified once per named class. The following two names are recognized:

combining Characters to form composite graphic symbols, such as characters listed in ISO/IEC 10646:1993 annex B.1.

combining_level3 Characters to form composite graphic symbols, that may also be represented by other characters, such as characters listed in ISO/IEC 10646-1:1993 annex B.2.

Note: The level3 does not exist in ISO/IEC 10646 anymore.

The class names "upper", "lower", "alpha", "digit", "space", "cntrl", "punct", "graph", "print", "xdigit", and "blank" are taken to mean the classes defined by the respective keywords.

width Define the column width of characters, for example for use of the C function wcwidth(). The operands are first a list for characters, possibly using various ellipses, and semicolon separated, then a <colon>, and then the width of these characters given as an unsigned positive integer. Such width-lists separated by <semicolon> may be given for the various widths. The default value of width of characters in class "cntrl" and class "combining" is 0, else the default value of width is 1. A width for a character may be overridden by a WIDTH specification in a charmap. This keyword is optional.

map Define the mapping of characters to other characters. The first operand is a string, defining the name of the mapping. The string only contains letters, digits and <hyphen-minus> and <low-line> from the portable character set. The following operands consist of character pairs, separated by semicolons. The characters in each character pair are separated by a comma and the pair enclosed by parentheses. The first character in each pair is the character to map from, the second the corresponding character to map to. This keyword is optional. The keyword can only be specified once per named mapping.

The mapping names "toupper", and "tolower" are taken to mean the mapping defined by the respective keywords.

Example of use of the "map" keyword:

map "kana",(<U30AB>,<U304B>);(<U30AC>,<U304C>);(<U30AD>,<U304D>)

This example introduces a new mapping "kana" that maps three Katakana characters to corresponding Hiragana characters.

Table 2 shows the allowed character class combinations.

Table 2: Valid Character Class Combinations

Class	upper	lower	alpha	digit	space	cntrl	punct	graph	print	xdigit	blank
upper		+	A	x	x	x	x	A	A	+	x
lower	+		A	x	x	x	x	A	A	+	x
alpha	+	+		x	x	x	x	A	A	+	x
digit	x	x	x		x	x	x	A	A	A	x
space	x	x	x	x		+	*	*	*	x	+
cntrl	x	x	x	x	+		x	x	x	x	+
punct	x	x	x	x	+	x		A	A	x	+
graph	+	+	+	+	+	x	+		A	+	+
print	+	+	+	+	+	x	+	+		+	+
xdigit	+	+	+	+	x	x	x	A	A		x
blank	x	x	x	x	A	+	*	*	*	x	

Note 1: Explanation of codes:
A Automatically included; see text
+ Permitted
x Mutually exclusive
* See note 2

Note 2: The <space> character, which is part of the "space" and "blank" class, cannot belong to "punct" or "graph", but automatically belong to the "print" class. Other "space" or "blank" characters can be classified as "punct", "graph", and/or "print".

4.3.2 Character string transliteration

The following keywords may be used to transliterate strings. The transliteration may for example be from the Cyrillic script to the Latin script. Transliteration is often language dependent, and the language to be transliterated to is identified with the FDCC-set, which may also be used to identify a specific language to be transliterated from. Transliteration of an incoming character string to a character string in a FDCC-set can be specified with the following keywords and transliteration statements.

translit_start The "translit_start" keyword is followed by one or more transliteration statements assigning character transliteration values to transliterating elements, and include statements copying transliteration specifications from other FDCC-sets.

translit_end The end of the transliteration statements.

For other keywords and transliteration statements, see clause 4.9 on LC_XLITERATE.

4.3.3 "i18n" LC_CTYPE category

The "i18n" FDCC-set for the LC_CTYPE is defined as follows:

```
LC_CTYPE
% The following is the ISO/IEC TR 30112 i18n fdcc-set LC_CTYPE category.
% It covers ISO/IEC 10646 collection 307 (Unicode version 5.0.0).
% The character classes and mapping tables were automatically generated
% using the gen-unicode-ctype.c program from the glibc project.
% This data reflects data from glibc version 2.14.1.
%
% The plan is to update this information to be aligned with a newer
% version of ISO/IEC 14651 and specification of the Unicode Standard.

% The "upper" class reflects the uppercase characters of class "alpha"
upper /
% BASIC LATIN/
   <U0041>..<U005A>;/
% LATIN-1 SUPPLEMENT/
   <U00C0>..<U00D6>;<U00D8>..<U00DE>;/
% LATIN EXTENDED-A/
   <U0100>..(2)..<U0136>;/
   <U0139>..(2)..<U0147>;/
   <U014A>..(2)..<U0178>;/
   <U0179>..(2)..<U017D>;/
% LATIN EXTENDED-B/
   <U0181>;<U0182>..(2)..<U0186>;<U0187>;/
   <U0189>..<U018B>;<U018E>..<U0191>;<U0193>;<U0194>;/
   <U0196>..<U0198>;<U019C>;<U019D>;<U019F>;/
   <U01A0>..(2)..<U01A4>;/
   <U01A6>;<U01A7>;<U01A9>;<U01AC>;<U01AE>;<U01AF>;<U01B1>..<U01B3>;/
   <U01B5>;<U01B7>;<U01B8>;<U01BC>;<U01C4>;<U01C5>;<U01C7>;<U01C8>;/
   <U01CA>;<U01CB>;/
   <U01CD>..(2)..<U01DB>;/
   <U01DE>..(2)..<U01EE>;/
   <U01F1>;<U01F2>;<U01F4>;<U01F6>..<U01F8>;<U01FA>..(2)..<U01FE>;/
   <U0200>..(2)..<U0232>;/
   <U023A>;<U023B>;<U023D>;<U023E>;/
   <U0241>;<U0243>..<U0246>;<U0248>;<U024A>;<U024C>;<U024E>;/
% BASIC GREEK/
   <U0370>;<U0372>;<U0376>;/
   <U0386>;<U0388>..<U038A>;<U038C>;<U038E>;<U038F>;<U0391>..<U03A1>;/
   <U03A3>..<U03AB>;<U03D8>..(2)..<U03DE>;/
% GREEK SYMBOLS AND COPTIC/
   <U03E0>..(2)..<U03EE>;<U03F4>;/
   <U03F7>;<U03F9>..<U03FA>;<U03FD>..<U03FF>;/
% CYRILLIC/
   <U0400>..<U042F>;<U0460>..(2)..<U047E>;/
   <U0480>;<U048A>..(2)..<U04BE>;<U04C0>;<U04C1>..(2)..<U04CD>;/
   <U04D0>..(2)..<U04FE>;/
% CYRILLIC SUPPLEMENT/
   <U0500>..(2)..<U0522>;/
% ARMENIAN/
   <U0531>..<U0556>;/
% GEORGIAN/
% is not addressed as the letters does not have a uppercase/lowercase relation/
% well, there are three georgian blocks defined; one caseless (the one usually/
% used), one defined as uppercase and one as lowercase. defining the uppercase one here/
   <U10A0>..<U10C5>;/
% LATIN EXTENDED ADDITIONAL/
   <U1E00>..(2)..<U1E7E>;/
   <U1E80>..(2)..<U1E94>;<U1E9E>;/
   <U1EA0>..(2)..<U1EFE>;/
% GREEK EXTENDED/
   <U1F08>..<U1F0F>;<U1F18>..<U1F1D>;<U1F28>..<U1F2F>;<U1F38>..<U1F3F>;/
   <U1F48>..<U1F4D>;<U1F59>..(2)..<U1F5F>;<U1F68>..<U1F6F>;/
   <U1F88>..<U1F8F>;<U1F98>..<U1F9F>;<U1FA8>..<U1FAF>;<U1FB8>..<U1FBC>;/
```



```
   <U1FC8>..<U1FCC>;<U1FD8>..<U1FDB>;<U1FE8>..<U1FEC>;<U1FF8>..<U1FFC>;/
% LETTERLIKE SYMBOLS/
   <U2126>;<U212A>..<U212B>;/
   <U2132>;/
% NUMBER FORMS/
   <U2160>..<U216F>;/
   <U2183>;/
% ENCLOSED ALPHANUMERICS/
   <U24B6>..<U24CF>;/
% GLAGOLITIC/
   <U2C00>..<U2C2E>;/
% LATIN EXTENDED-C/
   <U2C60>;<U2C62>..<U2C64>;<U2C67>..(2)..<U2C6B>;<U2C6D>..<U2C6F>;/
   <U2C72>;<U2C75>;<UA78B>;/
% COPTIC/
   <U2C80>..(2)..<U2CE2>;/
% CYRILLIC SUPPLEMENT 2/
   <UA640>..(2)..<UA65E>;<UA662>..(2)..<UA66C>;<UA680>..(2)..<UA696>;/
% LATIN EXTENDED-D/
   <UA722>..(2)..<UA72E>;<UA732>..(2)..<UA76E>;<UA779>..(2)..<UA77D>;/
   <UA77E>..(2)..<UA786>;/
% HALFWIDTH AND FULLWIDTH FORMS/
   <UFF21>..<UFF3A>;/
% DESERET/
   <U00010400>..<U00010427>

% The "lower" class reflects the lowercase characters of class "alpha"
lower /
% BASIC LATIN/
   <U0061>..<U007A>;/
% LATIN-1 SUPPLEMENT/
   <U00B5>;<U00DF>..<U00F6>;<U00F8>..<U00FF>;/
% LATIN EXTENDED-A/
   <U0101>..(2)..<U0137>;<U013A>..(2)..<U0148>;/
   <U014B>..(2)..<U0177>;<U017A>..(2)..<U017E>;<U017F>;/
% LATIN EXTENDED-B/
   <U0180>;<U0183>;<U0185>;<U0188>;<U018C>;<U0192>;<U0195>;/
   <U0199>;<U019A>;<U019E>;<U01A1>;<U01A3>;<U01A5>;<U01A8>;<U01AD>;/
   <U01B0>;<U01B4>;<U01B6>;<U01B9>;<U01BD>;<U01BF>;<U01C5>;<U01C6>;/
   <U01C8>;<U01C9>;<U01CB>;<U01CC>..(2)..<U01DC>;/
   <U01DD>..(2)..<U01EF>;<U01F2>;<U01F3>;<U01F5>;<U01F9>..(2)..<U01FF>;/
   <U0201>..(2)..<U021F>;<U0223>..(2)..<U0233>;/
   <U023C>;<U0242>;<U0247>..(2)..<U024F>;/
% IPA EXTENSIONS/
   <U0253>;<U0254>;<U0256>;<U0257>;<U0259>;<U025B>;<U0260>;<U0263>;<U0268>;/
   <U0269>;<U026B>;<U026F>;<U0272>;<U0275>;<U027D>;<U0280>;<U0283>;<U0288>..<U028C>;/
   <U0292>;/
% COMBINING DIACRITICAL MARKS/
   <U0345>;/
% BASIC GREEK/
   <U0371>;<U0373>;<U0377>;/
   <U037B>..<U037D>;/
   <U03AC>..<U03AF>;<U03B1>..<U03CE>;/
% GREEK SYMBOLS AND COPTIC/
   <U03D0>;<U03D1>;<U03D5>;<U03D6>;<U03D9>..(2)..<U03EF>;<U03F0>..<U03F2>;/
   <U03F5>;<U03F8>;<U03FB>;/
% CYRILLIC/
   <U0430>..<U045F>;<U0461>..(2)..<U047F>;/
   <U0481>;<U048B>..(2)..<U04BF>;<U04C2>..(2)..<U04CE>;/
   <U04CF>;/
   <U04D1>..(2)..<U0523>;/
% ARMENIAN/
   <U0561>..<U0586>;/
% PHONETIC EXTENSIONS/
   <U1D7D>;/
```



```
% LATIN EXTENDED ADDITIONAL/
   <U1E01>..(2)..<U1E95>;<U1E9B>..<U1E9D>;<U1E9F>;<U1EA1>..(2)..<U1EFF>;/
% GREEK EXTENDED/
   <U1F00>..<U1F07>;<U1F10>..<U1F15>;<U1F20>..<U1F27>;<U1F30>..<U1F37>;/
   <U1F40>..<U1F45>;<U1F51>..(2)..<U1F57>;<U1F60>..<U1F67>;<U1F70>..<U1F7D>;/
   <U1F80>..<U1F87>;<U1F90>..<U1F97>;<U1FA0>..<U1FA7>;<U1FB0>;<U1FB1>;/
   <U1FB3>;<U1FBE>;<U1FC3>;<U1FD0>;<U1FD1>;<U1FE0>;<U1FE1>;<U1FE5>;/
   <U1FF3>;/
% LETTERLIKE SYMBOLS/
   <U214E>;/
% NUMBER FORMS/
   <U2170>..<U217F>;<U2188>;/
% ENCLOSED ALPHANUMERICS/
   <U24D0>..<U24E9>;/
% GLAGOLITIC/
   <U2C30>..<U2C5E>;/
% LATIN EXTENDED-C/
   <U2C61>;<U2C65>;<U2C66>..(2)..<U2C6C>;<U2C71>;<U2C73>;<U2C74>;/
   <U2C76>..<U2C7A>;/
% COPTIC/
   <U2C81>..(2)..<U2CE3>;/
% GEORGIAN SUPPLEMENT/
% well, there are three georgian blocks defined; one caseless (the one usually/
% used), one defined as uppercase and one as lowercase. defining the lowercase one here/
   <U2D00>..<U2D25>;/
% CYRILLIC SUPPLEMENT 2/
   <UA641>..(2)..<UA65F>;<UA663>..(2)..<UA66D>;<UA681>..(2)..<UA697>;/
% LATIN EXTENDED-D/
   <UA723>..(2)..<UA72F>;<UA730>;<UA731>..(2)..<UA76F>;<UA771>..<UA778>;/
   <UA77A>..(2)..<UA77C>;<UA77F>..(2)..<UA787>;<UA78C>;/
% HALFWIDTH AND FULLWIDTH FORMS/
   <UFF41>..<UFF5A>;/
% DESERET/
   <U00010428>..<U0001044F>

% The "alpha" class of the "i18n" FDCC-set is reflecting
% the recommendations in TR 10176 annex A
alpha /
% BASIC LATIN/
   <U0041>..<U005A>;<U0061>..<U007A>;/
% LATIN-1 SUPPLEMENT/
   <U00AA>;<U00B5>;<U00BA>;<U00C0>..<U00D6>;<U00D8>..<U00F6>;/
   <U00F8>..<U00FF>;/
% LATIN EXTENDED-A/
   <U0100>..<U017F>;/
% LATIN EXTENDED-B/
   <U0180>..<U024F>;/
% IPA EXTENSIONS/
   <U0250>..<U02AF>;/
% SPACING MODIFIER LETTERS/
   <U02B0>..<U02C1>;<U02C6>..<U02D1>;<U02E0>..<U02E4>;/
   <U02EE>;/
% COMBINING DIACRITICAL MARKS/
   <U0345>;/
% BASIC GREEK/
   <U0370>..<U0373>;<U0376>..<U0377>;<U037A>..<U037D>;<U0386>;/
   <U0388>..<U038A>;<U038C>;<U038E>..<U03A1>;/
   <U03A3>..<U03CE>;/
% GREEK SYMBOLS AND COPTIC/
   <U03D0>..<U03F5>;<U03F7>..<U03FF>;/
% CYRILLIC/
   <U0400>..<U0481>;<U048A>..<U04FF>;/
% CYRILLIC SUPPLEMENT/
   <U0500>..<U0523>;/
% ARMENIAN/
```



```
   <U0531>..<U0556>;<U0559>;<U0561>..<U0587>;/
% HEBREW/
   <U05D0>..<U05EA>;<U05F0>..<U05F2>;/
% ARABIC/
   <U0621>..<U064A>;<U066E>..<U066F>;<U0671>..<U06D3>;/
   <U06D5>;<U06E5>..<U06E6>;<U06EE>..<U06EF>;<U06FA>..<U06FC>;<U06FF>;/
% SYRIAC/
   <U0710>;<U0712>..<U072F>;<U074D>..<U074F>;/
% ARABIC SUPPLEMENT/
   <U0750>..<U077F>;/
% THAANA/
   <U0780>..<U07A5>;<U07B1>;/
% NKO/
   <U07C0>..<U07EA>;<U07F4>..<U07F5>;<U07FA>;/
% - All Matras of Indic and Sinhala are moved from punct to alpha class/
% - Added Unicode 5.1 charctares of Indic scripts/
% DEVANAGARI/
   <U0901>..<U0939>;<U093C>..<U094D>;/
   <U0950>..<U0954>;<U0958>..<U0961>;/
   <U0962>;<U0963>;<U0972>;<U097B>..<U097F>;/
% TABLE 18 BENGALI/
   <U0981>..<U0983>;<U0985>..<U098C>;<U098F>;<U0990>;<U0993>..<U09A8>;/
   <U09AA>..<U09B0>;<U09B2>;<U09B6>..<U09B9>;<U09BC>..<U09C4>;/
   <U09C7>;<U09C8>;<U09CB>..<U09CE>;<U09D7>;/
   <U09DC>;<U09DD>;<U09DF>..<U09E3>;<U09F0>..<U09FA>;/
% GURMUKHI/
   <U0A01>..<U0A03>;<U0A05>..<U0A0A>;<U0A0F>;<U0A10>;<U0A13>..<U0A28>;/
   <U0A2A>..<U0A30>;<U0A32>;<U0A33>;<U0A35>;<U0A36>;<U0A38>;<U0A39>;/
   <U0A3C>;<U0A3E>..<U0A42>;<U0A47>;<U0A48>;<U0A4B>..<U0A4D>;<U0A51>;/
   <U0A59>..<U0A5C>;<U0A5E>;<U0A70>..<U0A75>;/
% GUJARATI/
   <U0A81>..<U0A83>;/
   <U0A85>..<U0A8D>;<U0A8F>..<U0A91>;<U0A93>..<U0AA8>;/
   <U0AAA>..<U0AB0>;<U0AB2>;<U0AB3>;<U0AB5>..<U0AB9>;<U0ABC>..<U0AC5>;/
   <U0AC7>..<U0AC9>;<U0ACB>..<U0ACD>;/
   <U0AD0>;<U0AE0>..<U0AE3>;<U0AF1>;/
% ORIYA/
   <U0B01>..<U0B03>;<U0B05>..<U0B0C>;<U0B0F>;<U0B10>;<U0B13>..<U0B28>;/
   <U0B2A>..<U0B30>;<U0B32>;<U0B33>;<U0B35>..<U0B39>;<U0B3C>..<U0B44>;/
   <U0B47>..<U0B48>;<U0B4B>..<U0B4D>;<U0B56>..<U0B57>;<U0B5C>;<U0B5D>;/
   <U0B5F>..<U0B63>;<U0B70>;<U0B71>;/
% TAMIL/
   <U0B82>;<U0B83>;<U0B85>..<U0B8A>;<U0B8E>..<U0B90>;<U0B92>..<U0B95>;<U0B99>;/
   <U0B9A>;<U0B9C>;<U0B9E>;<U0B9F>;<U0BA3>;<U0BA4>;<U0BA8>..<U0BAA>;/
   <U0BAE>..<U0BB9>;<U0BBE>..<U0BC2>;<U0BC6>..<U0BC8>;<U0BCA>..<U0BCD>;/
  <U0BD0>;<U0BD7>;<U0BF0>..<U0BFA>;/
% TELUGU/
   <U0C01>..<U0C03>;<U0C05>..<U0C0C>;<U0C0E>..<U0C10>;<U0C12>..<U0C28>;/
   <U0C2A>..<U0C33>;<U0C35>..<U0C39>;<U0C3D>..<U0C44>;<U0C46>..<U0C48>;/
   <U0C4A>..<U0C4D>;<U0C55>..<U0C56>;<U0C58>..<U0C59>;<U0C60>..<U0C63>;/
% KANNADA/
   <U0C82>..<U0C83>;<U0C85>..<U0C8C>;<U0C8E>..<U0C90>;<U0C92>..<U0CA8>;/
   <U0CAA>..<U0CB3>;<U0CB5>..<U0CB9>;<U0CBC>..<U0CC4>;<U0CC6>..<U0CC8>;<U0CCA>..<U0CCD>;/
   <U0CD5>..<U0CD6>;<U0CDE>;<U0CE0>..<U0CE3>;<U0CF1>;<U0CF2>;/
% MALAYALAM/
   <U0D02>..<U0D03>;<U0D05>..<U0D0C>;<U0D0E>..<U0D10>;<U0D12>..<U0D28>;/
   <U0D2A>..<U0D39>;<U0D3D>..<U0D44>;/
   <U0D46>..<U0D48>;<U0D4A>..<U0D4D>;<U0D57>;/
   <U0D60>..<U0D63>;<U0D79>..<U0D7F>;/
% SINHALA/
   <U0D82>..<U0D83>;<U0D85>..<U0D96>;<U0D9A>..<U0DB1>;<U0DB3>..<U0DBB>;<U0DBD>;/
   <U0DC0>..<U0DC6>;<U0DCA>;/
   <U0DCF>..<U0DD4>;<U0DD6>;<U0DD8>..<U0DDF>;<U0DF2>..<U0DF4>;/
% THAI/
   <U0E01>..<U0E2E>;<U0E30>..<U0E3A>;<U0E40>..<U0E45>;<U0E47>..<U0E4E>;/
```



```
% LAO/
   <U0E81>..<U0E82>;<U0E84>;<U0E87>..<U0E88>;<U0E8A>;<U0E8D>;/
   <U0E94>..<U0E97>;<U0E99>..<U0E9F>;<U0EA1>..<U0EA3>;<U0EA5>;<U0EA7>;/
   <U0EAA>..<U0EAB>;<U0EAD>..<U0EB0>;<U0EB2>..<U0EB3>;<U0EBD>;/
   <U0EC0>..<U0EC4>;<U0EC6>;<U0EDC>..<U0EDD>;/
% TIBETAN/
   <U0F00>;<U0F40>..<U0F47>;<U0F49>..<U0F6C>;<U0F88>..<U0F8B>;/
% MYANMAR/
   <U1000>..<U102A>;<U1050>..<U1055>;<U105A>..<U105D>;<U1061>;<U0165>;/
   <U1066>;<U106E>..<U1070>;<U1075>..<U1081>;<U108E>;/
% GEORGIAN/
   <U10A0>..<U10C5>;<U10D0>..<U10FA>;<U10FC>;/
% HANGUL JAMO/
   <U1100>..<U1159>;<U115F>..<U11A2>;<U11A8>..<U11F9>;/
% ETHIOPIC/
   <U1200>..<U1248>;<U124A>..<U124D>;/
   <U1250>..<U1256>;<U1258>;<U125A>..<U125D>;<U1260>..<U1288>;/
   <U128A>..<U128D>;<U1290>..<U12B0>;<U12B2>..<U12B5>;/
   <U12B8>..<U12BE>;<U12C0>;<U12C2>..<U12C5>;<U12C8>..<U12D6>;/
   <U12D8>..<U1310>;/
   <U1312>..<U1315>;<U1318>..<U135A>;/
% ETHIOPIC EXTENDED/
   <U1380>..<U138F>;/
% CHEROKEE/
   <U13A0>..<U13F4>;/
% UNIFIED CANADIAN ABORIGINAL SYLLABICS/
   <U1401>..<U166C>;<U166F>..<U1676>;/
% OGHAM/
   <U1681>..<U169A>;/
% RUNIC/
   <U16A0>..<U16EA>;<U16EE>..<U16F0>;/
% TAGALOG/
   <U1700>..<U170C>;<U170E>..<U1711>;/
% HANUNOO/
   <U1720>..<U1731>;/
% BUHID/
   <U1740>..<U1751>;/
% TAGBANWA/
   <U1760>..<U176C>;<U176E>..<U1770>;/
% KHMER/
   <U1780>..<U17B3>;<U17D7>;<U17DC>;/
% MONGOLIAN/
   <U1820>..<U1877>;<U1880>..<U18A8>;<U18AA>;/
% LIMBU/
   <U1900>..<U191C>;<U1946>..<U194F>;/
% TAI LE/
   <U1950>..<U196D>;<U1970>..<U1974>;/
% NEW TAI LUE/
   <U1980>..<U19A9>;<U19C1>..<U19C7>;<U19D0>..<U19D9>;/
% BUGINESE/
   <U1A00>..<U1A16>;/
% BALINESE/
   <U1B05>..<U1B33>;<U1B45>..<U1B4B>;<U1B50>..<U1B59>;/
% SUNDANESE/
   <U1B83>..<U1BA0>;<U1BAE>..<U1BAF>;/
% LEPCHA/
   <U1C00>..<U1C23>;<U1C4D>..<U1C4F>;/
% OL CHIKI/
   <U1C5A>..<U1C7D>;/
% PHONETIC EXTENSIONS/
   <U1D00>..<U1DBF>;/
% LATIN EXTENDED ADDITIONAL/
   <U1E00>..<U1E9F>;<U1EA0>..<U1EFF>;/
% GREEK EXTENDED/
   <U1F00>..<U1F15>;<U1F18>..<U1F1D>;<U1F20>..<U1F45>;<U1F48>..<U1F4D>;/
```



```
   <U1F50>..<U1F57>;<U1F59>;<U1F5B>;<U1F5D>;<U1F5F>..<U1F7D>;/
   <U1F80>..<U1FB4>;<U1FB6>..<U1FBC>;<U1FBE>;<U1FC2>..<U1FC4>;/
   <U1FC6>..<U1FCC>;<U1FD0>..<U1FD3>;<U1FD6>..<U1FDB>;<U1FE0>..<U1FEC>;/
   <U1FF2>..<U1FF4>;<U1FF6>..<U1FFC>;/
% SUPERSCRIPTS AND SUBSCRIPTS/
   <U2071>;<U207F>;<U2090>..<U2094>;/
% LETTERLIKE SYMBOLS/
   <U2102>;<U2107>;<U210A>..<U2113>;<U2115>;<U2119>..<U211D>;<U2124>;/
   <U2126>;<U2128>..<U212D>;<U212F>..<U2139>;/
   <U213C>..<U213F>;<U2145>..<U2149>;<U214E>;/
% NUMBER FORMS/
   <U2160>..<U2188>;/
% ENCLOSED ALPHANUMERICS/
   <U249C>..<U24E9>;/
% GLAGOLITIC/
   <U2C00>..<U2C2E>;<U2C30>..<U2C5E>;/
% LATIN EXTENDED-C/
   <U2C60>..<U2C6F>;<U2C71>..<U2C7D>;/
% COPTIC/
   <U2C80>..<U2CE4>;/
% GEORGIAN SUPPLEMENT/
   <U2D00>..<U2D25>;/
% TIFINAGH/
   <U2D30>..<U2D65>;<U2D6F>;/
% ETHIOPIC EXTENDED/
   <U2D80>..<U2D96>;<U2DA0>..<U2DA6>;<U2DA8>..<U2DAE>;<U2DB0>..<U2DB6>;/
   <U2DB8>..<U2DBE>;<U2DC0>..<U2DC6>;<U2DC8>..<U2DCE>;<U2DD0>..<U2DD6>;/
   <U2DD8>..<U2DDE>;/
% CJK SYMBOLS AND PUNCTUATION/
   <U3005>..<U3007>;<U3021>..<U3029>;<U3031>..<U3035>;<U3038>..<U303C>;/
% HIRAGANA/
   <U3041>..<U3096>;<U309D>..<U309F>;/
% KATAKANA/
   <U30A1>..<U30FA>;<U30FC>..<U30FF>;/
% BOPOMOFO/
   <U3105>..<U312D>;/
% HANGUL COMPATIBILITY JAMO/
   <U3131>..<U318E>;/
% BOPOMOFO EXTENDED/
   <U31A0>..<U31B7>;/
% KATAKANA PHONETIC EXTENSIONS/
   <U31F0>..<U31FF>;/
% CJK UNIFIED IDEOGRAPHS EXTENSION/
   <U3400>..<U4DB5>;/
% CJK UNIFIED IDEOGRAPHS/
   <U4E00>..<U9FBB>;/
% YI SYLLABLES/
   <UA000>..<UA48C>;/
% VAI SYLLABLES/
   <UA500>..<UA60B>;<UA610>..<UA61F>;<UA62A>..<UA62B>;/
% CYRILLIC SUPPLEMENT 2/
   <UA640>..<UA65F>;<UA662>..<UA66E>;<UA680>..<UA697>;/
% LATIN EXTENDED-D/
   <UA717>..<UA71F>;<UA722>..<UA78C>;<UA7FB>..<UA7FF>;/
% SYLOTI NEGRI/
   <UA800>;<UA801>;<UA803>..<UA805>;<UA807>..<UA80A>;<UA80C>..<UA822>;/
% PHAGS PA/
   <UA840>..<UA873>;/
% SAURASHTRA/
   <UA882>..<UA8B3>;/
% KAYAH LI/
   <UA90A>..<UA92D>;/
% REJANG/
   <UA930>..<UA946>;/
% CHAM/
```



```
   <UAA00>..<UAA28>;<UAA40>..<UAA42>;<UAA44>..<UAA4B>;/
% HANGUL SYLLABLES/
   <UAC00>..<UD7A3>;/
% CJK COMPATIBILITY IDEOGRAPHS/
   <UF900>..<UFA2D>;<UFA30>..<UFA6A>;/
   <UFA70>..<UFAD9>;/
% ALPHABETIC PRESENTATION FORMS/
   <UFB00>..<UFB06>;<UFB13>..<UFB17>;<UFB1D>;<UFB1F>..<UFB28>;/
   <UFB2A>..<UFB36>;<UFB38>..<UFB3C>;<UFB3E>;<UFB40>;<UFB41>;<UFB43>;/
   <UFB44>;<UFB46>..<UFB4F>;/
% ARABIC PRESENTATION FORMS-A/
   <UFB50>..<UFBB1>;<UFBD3>..<UFD3D>;<UFD50>..<UFD8F>;<UFD92>..<UFDC7>;/
   <UFDF0>..<UFDFB>;/
% ARABIC PRESENTATION FORMS-B/
   <UFE70>..<UFE74>;<UFE76>..<UFEFC>;/
% HALFWIDTH AND FULLWIDTH FORMS/
   <UFF21>..<UFF3A>;<UFF41>..<UFF5A>;<UFF66>..<UFFBE>;<UFFC2>..<UFFC7>;/
   <UFFCA>..<UFFCF>;<UFFD2>..<UFFD7>;<UFFDA>..<UFFDC>;/
% LINEAR B SYLLABARY/
   <U00010000>..<U0001000B>;<U0001000D>..<U00010026>;/
   <U00010028>..<U0001003A>;<U0001003C>..<U0001003D>;/
   <U0001003F>..<U0001004D>;<U00010050>..<U0001005D>;/
% LINEAR B IDEOGRAMS/
   <U00010080>..<U000100FA>;/
% ANCIENT GREEK NUMBERS/
   <U00010140>..<U00010174>;/
% LYCIAN/
   <U00010280>..<U0001029C>;/
% CARIAN/
   <U000102A0>..<U000102D0>;/
% OLD ITALIC/
   <U00010300>..<U0001031E>;/
% GOTHIC/
   <U00010330>..<U0001034A>;/
% UGARITIC/
   <U00010380>..<U0001039D>;/
% OLD PERSIAN/
   <U000103A0>..<U000103C3>;<U000103C8>..<U000103CF>;/
   <U000103D1>..<U000103D5>;/
% DESERET/
   <U00010400>..<U0001044F>;/
% SHAVIAN/
   <U00010450>..<U0001047F>;/
% OSMANYA/
   <U00010480>..<U0001049D>;<U000104A0>..<U000104A9>;/
% CYPRIOT SYLLABARY/
   <U00010800>..<U00010805>;<U00010808>;<U0001080A>..<U00010835>;/
   <U00010837>..<U00010838>;<U0001083C>;<U0001083F>;/
% PHOENICIAN/
   <U00010900>..<U00010915>;<U00010A00>;<U00010A10>..<U00010A13>;/
% KHAROSHTI/
   <U00010A15>..<U00010A17>;<U00010A19>..<U00010A33>;/
% CUNEIFORM/
   <U00012000>..<U0001236E>;/
% CUNEIFORM NUMBERS AND PONCTUATION/
   <U00012400>..<U00012462>;/
% BYZANTINE MUSICAL SYMBOLS/
% MATHEMATICAL ALPHANUMERIC SYMBOLS/
   <U0001D400>..<U0001D454>;<U0001D456>..<U0001D49C>;/
   <U0001D49E>..<U0001D49F>;<U0001D4A2>;<U0001D4A5>..<U0001D4A6>;/
   <U0001D4A9>..<U0001D4AC>;<U0001D4AE>..<U0001D4B9>;<U0001D4BB>;/
   <U0001D4BD>..<U0001D4C3>;<U0001D4C5>..<U0001D505>;/
   <U0001D507>..<U0001D50A>;<U0001D50D>..<U0001D514>;/
   <U0001D516>..<U0001D51C>;<U0001D51E>..<U0001D539>;/
   <U0001D53B>..<U0001D53E>;<U0001D540>..<U0001D544>;<U0001D546>;/
```



```
   <U0001D54A>..<U0001D550>;<U0001D552>..<U0001D6A5>;/
   <U0001D6A8>..<U0001D6C0>;<U0001D6C2>..<U0001D6DA>;/
   <U0001D6DC>..<U0001D6FA>;<U0001D6FC>..<U0001D714>;/
   <U0001D716>..<U0001D734>;<U0001D736>..<U0001D74E>;/
   <U0001D750>..<U0001D76E>;<U0001D770>..<U0001D788>;/
   <U0001D78A>..<U0001D7A8>;<U0001D7AA>..<U0001D7C2>;/
   <U0001D7C4>..<U0001D7CB>;<U0001D7CE>..<U0001D7FF>;/
% CJK UNIFIED IDEOGRAPHS EXTENSION/
   <U00020000>..<U0002A6D6>;/
% CJK COMPATIBILITY IDEOGRAPHS SUPPLEMENT/
   <U0002F800>..<U0002FA1D>;/
% The non-ASCII number characters are included here because ISO C 99    /
% forbids us to classify them as digits; however, they behave more like /
% alphanumeric than like punctuation.                                   /
% ARABIC/
   <U0660>..<U0669>;<U06F0>..<U06F9>;/
% DEVANAGARI/
   <U0966>..<U096F>;/
% BENGALI/
   <U09E6>..<U09EF>;/
% GURMUKHI/
   <U0A66>..<U0A6F>;/
% GUJARATI/
   <U0AE6>..<U0AEF>;/
% ORIYA/
   <U0B66>..<U0B6F>;/
% TAMIL/
   <U0BE6>..<U0BEF>;/
% TELUGU/
   <U0C66>..<U0C6F>;<U0C78>..<U0C7F>;/
% KANNADA/
   <U0CE6>..<U0CEF>;/
% MALAYALAM/
   <U0D66>..<U0D75>;<U0D70>..<U0D75>;/
% THAI/
   <U0E50>..<U0E59>;/
% LAO/
   <U0ED0>..<U0ED9>;/
% TIBETAN/
   <U0F20>..<U0F29>;/
% MYANMAR/
   <U1040>..<U1049>;/
% KHMER/
   <U17E0>..<U17E9>;/
% MONGOLIAN/
   <U1810>..<U1819>;/
% SUNDANESE/
   <U1BB0>..<U1BB9>;/
% LEPCHA/
   <U1C40>..<U1C49>;/
% OL CHIKI/
   <U1C50>..<U1C59>;/
% VAI/
   <UA620>..<UA629>;/
% SAURASHTRA/
   <UA8D0>..<UA8D9>;/
% KAYAH LI/
   <UA900>..<UA909>;/
% CHAM/
   <UAA50>..<UAA59>;/
% HALFWIDTH AND FULLWIDTH FORMS/
   <UFF10>..<UFF19>

% The "digit" class must only contain the BASIC LATIN digits, says ISO C 99
% (sections 7.25.2.1.5 and 5.2.1).
```



```
digit /
   <U0030>..<U0039>

% The "outdigit" information is by default "0" to "9".  We don't have to
% provide it here since localedef will fill in the bits and it would
% prevent locales copy-ing this file define their own values.
% outdigit <U0030>..<U0039>

space /
% ISO/IEC 6429/
   <U0009>..<U000D>;/
% BASIC LATIN/
   <U0020>;/
% OGHAM/
   <U1680>;/
% MONGOL/
   <U180E>;/
% GENERAL PUNCTUATION/
   <U2000>..<U2006>;<U2008>..<U200A>;<U2028>;<U2029>;<U205F>;/
% CJK SYMBOLS AND PUNCTUATION, HIRAGANA/
   <U3000>

cntrl /
   <U0000>..<U001F>;<U007F>..<U009F>;/
% Treat the Line/Paragraph Separators as control characters, like Line Feed./
   <U2028>;<U2029>

punct /
   <U0021>..<U002F>;<U003A>..<U0040>;<U005B>..<U0060>;<U007B>..<U007E>;/
   <U00A0>..<U00A9>;<U00AB>..<U00B4>;<U00B6>..<U00B9>;<U00BB>..<U00BF>;/
   <U00D7>;<U00F7>;<U02C2>..<U02C5>;<U02D2>..<U02DF>;<U02E5>..<U02ED>;/
   <U02EF>..<U0344>;<U0346>..<U036F>;<U0374>..<U0375>;<U037E>;/
   <U0384>..<U0385>;<U0387>;<U03F6>;<U0482>..<U0486>;<U0488>..<U0489>;/
   <U055A>..<U055F>;<U0589>..<U058A>;<U0591>..<U05C7>;<U05F3>..<U05F4>;/
   <U0600>..<U0603>;<U060B>..<U061B>;<U061E>..<U061F>;/
   <U064B>..<U065E>;<U066A>..<U066D>;<U0670>;<U06D4>;<U06D6>..<U06E4>;/
   <U06E7>..<U06ED>;<U06FD>..<U06FE>;<U0700>..<U070D>;<U070F>;<U0711>;/
   <U0730>..<U074A>;<U07A6>..<U07B0>;<U07EB>..<U07F3>;<U07F6>..<U07F9>;/
   <U0964>;<U0965>;/
   <U0E2F>;/
   <U0E3F>;<U0E46>;<U0E4F>;<U0E5A>..<U0E5B>;<U0EB1>;<U0EB4>..<U0EB9>;/
   <U0EBB>..<U0EBC>;<U0EC8>..<U0ECD>;<U0F01>..<U0F1F>;<U0F2A>..<U0F3F>;/
   <U0F71>..<U0F87>;<U0F90>..<U0F97>;<U0F99>..<U0FBC>;<U0FBE>..<U0FCC>;/
   <U0FCE>..<U0FD4>;<U102B>..<U103F>;<U104A>..<U104F>;<U1056>..<U1059>;/
   <U105E>..<U1060>;<U1062>..<U1064>;<U1067>..<U106D>;<U1071>..<U1074>;/
   <U1082>..<U108D>;<U108F>..<U1099>;<U109E>;<U109F>;/
   <U10FB>;<U135F>..<U137C>;<U1390>..<U1399>;/
   <U166D>..<U166E>;<U169B>..<U169C>;<U16EB>..<U16ED>;<U1712>..<U1714>;/
   <U1732>..<U1736>;<U1752>..<U1753>;<U1772>..<U1773>;<U17B4>..<U17D6>;/
   <U17D8>..<U17DB>;<U17DD>;<U17F0>..<U17F9>;<U1800>..<U180D>;<U18A9>;/
   <U1920>..<U192B>;<U1930>..<U193B>;<U1940>;<U1944>..<U1945>;/
   <U19B0>..<U19C0>;<U19C8>..<U19C9>;<U19DE>..<U19FF>;<U1A17>..<U1A1B>;/
   <U1A1E>..<U1A1F>;<U1B00>..<U1B04>;<U1B34>..<U1B44>;<U1B5A>..<U1B7C>;/
   <U1B80>..<U1B82>;<U1BA1>..<U1BAA>;<U1C24>..<U1C37>;<U1C3B>..<U1C3F>;/
   <U1C7E>..<U1C7F>;/
   <U1DC0>..<U1DE6>;<U1DFE>..<U1DFF>;<U1FBD>;<U1FBF>..<U1FC1>;/
   <U1FCD>..<U1FCF>;<U1FDD>..<U1FDF>;<U1FED>..<U1FEF>;<U1FFD>..<U1FFE>;/
   <U2007>;<U200B>..<U2027>;<U202A>..<U205E>;<U2060>..<U2064>;/
   <U206A>..<U2070>;<U2074>..<U207E>;<U2080>..<U208E>;<U20A0>..<U20B5>;/
   <U20D0>..<U20F0>;<U2100>..<U2101>;<U2103>..<U2106>;<U2108>..<U2109>;/
   <U2114>;<U2116>..<U2118>;<U211E>..<U2123>;<U2125>;<U2127>;<U212E>;/
   <U213A>..<U213B>;<U2140>..<U2144>;<U214A>..<U214D>;<U2153>..<U215F>;/
   <U2190>..<U23E7>;<U2400>..<U2426>;<U2440>..<U244A>;<U2460>..<U249B>;/
   <U24EA>..<U269D>;<U26A0>..<U26C3>;<U2701>..<U2704>;<U2706>..<U2709>;/
   <U270C>..<U2727>;<U2729>..<U274B>;<U274D>;<U274F>..<U2752>;<U2756>;/
```



```
   <U2758>..<U275E>;<U2761>..<U2794>;<U2798>..<U27AF>;<U27B1>..<U27BE>;/
   <U27C0>..<U27CA>;<U27CC>;<U27D0>..<U27EF>;<U27F0>..<U2B4C>;/
   <U2B50>..<U2B54>;<U2DE0>..<U2DFF>;/
   <U2CE5>..<U2CEA>;<U2CF9>..<U2CFF>;<U2E00>..<U2E30>;/
   <U2E80>..<U2E99>;<U2E9B>..<U2EF3>;<U2F00>..<U2FD5>;<U2FF0>..<U2FFB>;/
   <U3001>..<U3004>;<U3008>..<U3020>;<U302A>..<U3030>;<U3036>..<U3037>;/
   <U303D>..<U303F>;<U3099>..<U309C>;<U30A0>;<U30FB>;<U3190>..<U319F>;/
   <U31C0>..<U31CF>;<U3200>..<U321E>;<U3220>..<U3243>;<U3250>..<U32FE>;/
   <U3300>..<U33FF>;<U4DC0>..<U4DFF>;<UA490>..<UA4C6>;<UA60C>..<UA60F>;/
   <UA66F>..<UA673>;<UA67C>..<UA67F>;<UA700>..<UA716>;/
   <UA720>..<UA721>;<UA802>;<UA806>;<UA80B>;<UA823>..<UA82B>;/
   <UA874>..<UA877>;<UA880>;<UA881>;<UA8B4>..<UA8C4>;<UA8CE>..<UA8CF>;/
   <UA92E>..<UA92F>;<UA947>..<UA953>;<UA95F>;<UAA29>..<UAA36>;<UAA43>;/
   <UAA4C>..<UAA4D>;<UAA5C>..<UAA5F>;/
   <UE000>..<UF8FF>;<UFB1E>;<UFB29>;<UFD3E>..<UFD3F>;/
   <UFDFC>..<UFDFD>;<UFE00>..<UFE19>;<UFE20>..<UFE26>;<UFE30>..<UFE52>;/
   <UFE54>..<UFE66>;<UFE68>..<UFE6B>;<UFEFF>;<UFF01>..<UFF0F>;/
   <UFF1A>..<UFF20>;<UFF3B>..<UFF40>;<UFF5B>..<UFF65>;<UFFE0>..<UFFE6>;/
   <UFFE8>..<UFFEE>;<UFFF9>..<UFFFD>;<U00010100>..<U00010102>;/
   <U00010107>..<U00010133>;<U00010137>..<U0001013F>;/
   <U00010175>..<U0001018A>;<U00010320>..<U00010323>;<U0001039F>;/
   <U000103D0>;<U00010916>..<U00010919>;<U0001091F>;/
   <U00010A01>..<U00010A03>;<U00010A05>..<U00010A06>;/
   <U00010A0C>..<U00010A0F>;<U00010A38>..<U00010A3A>;/
   <U00010A3F>..<U00010A47>;<U00010A50>..<U00010A58>;/
   <U00012470>..<U00012473>;<U0001D000>..<U0001D0F5>;/
   <U0001D100>..<U0001D126>;<U0001D129>..<U0001D1DD>;/
   <U0001D200>..<U0001D245>;<U0001D300>..<U0001D356>;/
   <U0001D360>..<U0001D371>;<U0001D6C1>;<U0001D6DB>;<U0001D6FB>;<U0001D715>;/
   <U0001D735>;<U0001D74F>;<U0001D76F>;<U0001D789>;<U0001D7A9>;<U0001D7C3>;/
   <U000E0001>;<U000E0020>..<U000E007F>;<U000E0100>..<U000E01EF>;/
   <U000F0000>..<U000FFFFD>;<U00100000>..<U0010FFFD>

graph /
   <U0021>..<U007E>;<U00A0>..<U0377>;<U037A>..<U037E>;/
   <U0384>..<U038A>;<U038C>;<U038E>..<U03A1>;<U03A3>..<U0486>;/
   <U0488>..<U0523>;<U0531>..<U0556>;<U0559>..<U055F>;/
   <U0561>..<U0587>;<U0589>..<U058A>;<U0591>..<U05C7>;<U05D0>..<U05EA>;/
   <U05F0>..<U05F4>;<U0600>..<U0603>;<U0606>..<U061B>;/
   <U061E>..<U061F>;<U0621>..<U065E>;<U0660>..<U070D>;/
   <U070F>..<U074A>;<U074D>..<U07B1>;<U07C0>..<U07FA>;/
   <U0901>..<U0939>;<U093C>..<U094D>;<U0950>..<U0954>;<U0958>..<U0972>;/
   <U097B>..<U097F>;<U0981>..<U0983>;<U0985>..<U098C>;<U098F>..<U0990>;/
   <U0993>..<U09A8>;<U09AA>..<U09B0>;<U09B2>;<U09B6>..<U09B9>;/
   <U09BC>..<U09C4>;<U09C7>..<U09C8>;<U09CB>..<U09CE>;<U09D7>;/
   <U09DC>..<U09DD>;<U09DF>..<U09E3>;<U09E6>..<U09FA>;<U0A01>..<U0A03>;/
   <U0A05>..<U0A0A>;<U0A0F>..<U0A10>;<U0A13>..<U0A28>;<U0A2A>..<U0A30>;/
   <U0A32>..<U0A33>;<U0A35>..<U0A36>;<U0A38>..<U0A39>;<U0A3C>;/
   <U0A3E>..<U0A42>;<U0A47>..<U0A48>;<U0A4B>..<U0A4D>;<U0A51>;/
   <U0A59>..<U0A5C>;/
   <U0A5E>;<U0A66>..<U0A75>;<U0A81>..<U0A83>;<U0A85>..<U0A8D>;/
   <U0A8F>..<U0A91>;<U0A93>..<U0AA8>;<U0AAA>..<U0AB0>;<U0AB2>..<U0AB3>;/
   <U0AB5>..<U0AB9>;<U0ABC>..<U0AC5>;<U0AC7>..<U0AC9>;<U0ACB>..<U0ACD>;/
   <U0AD0>;<U0AE0>..<U0AE3>;<U0AE6>..<U0AEF>;<U0AF1>;<U0B01>..<U0B03>;/
   <U0B05>..<U0B0C>;<U0B0F>..<U0B10>;<U0B13>..<U0B28>;<U0B2A>..<U0B30>;/
   <U0B32>..<U0B33>;<U0B35>..<U0B39>;<U0B3C>..<U0B44>;<U0B47>..<U0B48>;/
   <U0B4B>..<U0B4D>;<U0B56>..<U0B57>;<U0B5C>..<U0B5D>;<U0B5F>..<U0B63>;/
   <U0B66>..<U0B71>;<U0B82>..<U0B83>;<U0B85>..<U0B8A>;<U0B8E>..<U0B90>;/
   <U0B92>..<U0B95>;<U0B99>..<U0B9A>;<U0B9C>;<U0B9E>..<U0B9F>;/
   <U0BA3>..<U0BA4>;<U0BA8>..<U0BAA>;<U0BAE>..<U0BB9>;<U0BBE>..<U0BC2>;/
   <U0BC6>..<U0BC8>;<U0BCA>..<U0BCD>;<U0BD0>;<U0BD7>;<U0BE6>..<U0BFA>;/
   <U0C01>..<U0C03>;<U0C05>..<U0C0C>;<U0C0E>..<U0C10>;<U0C12>..<U0C28>;/
   <U0C2A>..<U0C33>;<U0C35>..<U0C39>;<U0C3D>..<U0C44>;<U0C46>..<U0C48>;/
   <U0C4A>..<U0C4D>;<U0C55>..<U0C56>;<U0C58>..<U0C59>;<U0C60>..<U0C63>;/
   <U0C66>..<U0C6F>;<U0C78>..<U0C7F>;/
```


<U0C82>..<U0C83>;<U0C85>..<U0C8C>;<U0C8E>..<U0C90>;<U0C92>..<U0CA8>;/
<U0CAA>..<U0CB3>;<U0CB5>..<U0CB9>;<U0CBC>..<U0CC4>;<U0CC6>..<U0CC8>;/
<U0CCA>..<U0CCD>;<U0CD5>..<U0CD6>;<U0CDE>;<U0CE0>..<U0CE3>;/
<U0CE6>..<U0CEF>;<U0CF1>..<U0CF2>;<U0D02>..<U0D03>;<U0D05>..<U0D0C>;/
<U0D0E>..<U0D10>;<U0D12>..<U0D28>;<U0D2A>..<U0D39>;<U0D3D>..<U0D44>;/
<U0D46>..<U0D48>;<U0D4A>..<U0D4D>;<U0D57>;<U0D60>..<U0D63>;/
<U0D66>..<U0D75>;<U0D79>..<U0D7F>;<U0D82>..<U0D83>;<U0D85>..<U0D96>;/
<U0D9A>..<U0DB1>;/
<U0DB3>..<U0DBB>;<U0DBD>;<U0DC0>..<U0DC6>;<U0DCA>;<U0DCF>..<U0DD4>;/
<U0DD6>;<U0DD8>..<U0DDF>;<U0DF2>..<U0DF4>;<U0E01>..<U0E3A>;/
<U0E3F>..<U0E5B>;<U0E81>..<U0E82>;<U0E84>;<U0E87>..<U0E88>;<U0E8A>;/
<U0E8D>;<U0E94>..<U0E97>;<U0E99>..<U0E9F>;<U0EA1>..<U0EA3>;<U0EA5>;/
<U0EA7>;<U0EAA>..<U0EAB>;<U0EAD>..<U0EB9>;<U0EBB>..<U0EBD>;/
<U0EC0>..<U0EC4>;<U0EC6>;<U0EC8>..<U0ECD>;<U0ED0>..<U0ED9>;/
<U0EDC>..<U0EDD>;<U0F00>..<U0F47>;<U0F49>..<U0F6C>;<U0F71>..<U0F8B>;/
<U0F90>..<U0F97>;<U0F99>..<U0FBC>;<U0FBE>..<U0FCC>;<U0FCE>..<U0FD4>;/
<U1000>..<U102A>;<U102B>..<U103F>;<U1040>..<U1099>;<U109E>;<U109F>;/
<U10A0>..<U10C5>;<U10D0>..<U10FC>;/
<U1100>..<U1159>;<U115F>..<U11A2>;<U11A8>..<U11F9>;<U1200>..<U1248>;/
<U124A>..<U124D>;<U1250>..<U1256>;<U1258>;<U125A>..<U125D>;/
<U1260>..<U1288>;<U128A>..<U128D>;<U1290>..<U12B0>;<U12B2>..<U12B5>;/
<U12B8>..<U12BE>;<U12C0>;<U12C2>..<U12C5>;<U12C8>..<U12D6>;/
<U12D8>..<U1310>;<U1312>..<U1315>;<U1318>..<U135A>;<U135F>..<U137C>;/
<U1380>..<U1399>;<U13A0>..<U13F4>;<U1401>..<U1676>;<U1681>..<U169C>;/
<U16A0>..<U16F0>;<U1700>..<U170C>;<U170E>..<U1714>;<U1720>..<U1736>;/
<U1740>..<U1753>;<U1760>..<U176C>;<U176E>..<U1770>;<U1772>..<U1773>;/
<U1780>..<U17DD>;<U17E0>..<U17E9>;<U17F0>..<U17F9>;<U1800>..<U180D>;/
<U1810>..<U1819>;<U1820>..<U1877>;<U1880>..<U18AA>;<U1900>..<U191C>;/
<U1920>..<U192B>;<U1930>..<U193B>;<U1940>;<U1944>..<U196D>;/
<U1970>..<U1974>;<U1980>..<U19A9>;<U19B0>..<U19C9>;<U19D0>..<U19D9>;/
<U19DE>..<U1A1B>;<U1A1E>..<U1A1F>;<U1B00>..<U1B4B>;<U1B50>..<U1B7C>;/
<U1B80>..<U1BAA>;<U1BAE>..<U1BB9>;<U1C00>..<U1C37>;<U1C3B>..<U1C49>;/
<U1C4D>..<U1C7F>;/
<U1D00>..<U1DE6>;<U1DFE>..<U1E9F>;<U1EA0>..<U1EFF>;<U1F00>..<U1F15>;/
<U1F18>..<U1F1D>;<U1F20>..<U1F45>;<U1F48>..<U1F4D>;<U1F50>..<U1F57>;/
<U1F59>;<U1F5B>;<U1F5D>;<U1F5F>..<U1F7D>;<U1F80>..<U1FB4>;/
<U1FB6>..<U1FC4>;<U1FC6>..<U1FD3>;<U1FD6>..<U1FDB>;<U1FDD>..<U1FEF>;/
<U1FF2>..<U1FF4>;<U1FF6>..<U1FFE>;<U2007>;<U200B>..<U2027>;/
<U202A>..<U205E>;<U2060>..<U2064>;<U206A>..<U2071>;<U2074>..<U208E>;/
<U2090>..<U2094>;<U20A0>..<U20B5>;<U20D0>..<U20F0>;<U2100>..<U214F>;/
<U2153>..<U2188>;<U2190>..<U23E7>;<U2400>..<U2426>;<U2440>..<U244A>;/
<U2460>..<U269D>;<U26A0>..<U26C3>;<U2701>..<U2704>;<U2706>..<U2709>;/
<U270C>..<U2727>;<U2729>..<U274B>;<U274D>;<U274F>..<U2752>;<U2756>;/
<U2758>..<U275E>;<U2761>..<U2794>;<U2798>..<U27AF>;<U27B1>..<U27BE>;/
<U27C0>..<U27CA>;<U27CC>;<U27D0>..<U27EF>;<U27F0>..<U2B4C>;/
<U2B50>..<U2B54>;/
<U2C00>..<U2C2E>;<U2C30>..<U2C5E>;<U2C60>..<U2C6F>;<U2C71>..<U2C7D>;/
<U2C80>..<U2CEA>;<U2CF9>..<U2D25>;<U2D30>..<U2D65>;<U2D6F>;/
<U2D80>..<U2D96>;<U2DA0>..<U2DA6>;<U2DA8>..<U2DAE>;<U2DB0>..<U2DB6>;/
<U2DB8>..<U2DBE>;<U2DC0>..<U2DC6>;<U2DC8>..<U2DCE>;<U2DD0>..<U2DD6>;/
<U2DD8>..<U2DDE>;<U2DE0>..<U2E30>;<U2E80>..<U2E99>;/
<U2E9B>..<U2EF3>;<U2F00>..<U2FD5>;<U2FF0>..<U2FFB>;<U3001>..<U303F>;/
<U3041>..<U3096>;<U3099>..<U30FF>;<U3105>..<U312D>;<U3131>..<U318E>;/
<U3190>..<U31B7>;<U31C0>..<U31CF>;<U31F0>..<U321E>;<U3220>..<U3243>;/
<U3250>..<U32FE>;<U3300>..<U4DB5>;<U4DC0>..<U9FBB>;<UA000>..<UA48C>;/
<UA490>..<UA4C6>;<UA500>..<UA62B>;<UA640>..<UA65F>;<UA662>..<UA673>;/
<UA67C>..<UA697>;<UA700>..<UA78C>;<UA7FB>..<UA7FF>;<UA800>..<UA82B>;/
<UA840>..<UA877>;<UA880>..<UA8C4>;<UA8CE>..<UA8D9>;<UA900>..<UA953>;/
<UA95F>;<UAA00>..<UAA36>;<UAA40>..<UAA4D>;<UAA50>..<UAA59>;/
<UAA5C>..<UAA5F>;<UAC00>..<UD7A3>;<UE000>..<UFA2D>;<UFA30>..<UFA6A>;/
<UFA70>..<UFAD9>;<UFB00>..<UFB06>;<UFB13>..<UFB17>;<UFB1D>..<UFB36>;/
<UFB38>..<UFB3C>;<UFB3E>;<UFB40>..<UFB41>;<UFB43>..<UFB44>;/
<UFB46>..<UFBB1>;<UFBD3>..<UFD3F>;<UFD50>..<UFD8F>;<UFD92>..<UFDC7>;/
<UFDF0>..<UFDFD>;<UFE00>..<UFE19>;<UFE20>..<UFE26>;<UFE30>..<UFE52>;/
<UFE54>..<UFE66>;<UFE68>..<UFE6B>;<UFE70>..<UFE74>;<UFE76>..<UFEFC>;/


```
    <UFEFF>;<UFF01>..<UFFBE>;<UFFC2>..<UFFC7>;<UFFCA>..<UFFCF>;/
    <UFFD2>..<UFFD7>;<UFFDA>..<UFFDC>;<UFFE0>..<UFFE6>;<UFFE8>..<UFFEE>;/
    <UFFF9>..<UFFFD>;<U00010000>..<U0001000B>;<U0001000D>..<U00010026>;/
    <U00010028>..<U0001003A>;<U0001003C>..<U0001003D>;/
    <U0001003F>..<U0001004D>;<U00010050>..<U0001005D>;/
    <U00010080>..<U000100FA>;<U00010100>..<U00010102>;/
    <U00010107>..<U00010133>;<U00010137>..<U0001018A>;/
    <U00010190>..<U0001019B>;<U000101D0>..<U000101FD>;/
    <U00010280>..<U0001029C>;<U000102A0>..<U000102D0>;/
    <U00010300>..<U0001031E>;<U00010320>..<U00010323>;/
    <U00010330>..<U0001034A>;<U00010380>..<U0001039D>;/
    <U0001039F>..<U000103C3>;<U000103C8>..<U000103D5>;/
    <U00010400>..<U0001049D>;<U000104A0>..<U000104A9>;/
    <U00010800>..<U00010805>;<U00010808>;<U0001080A>..<U00010835>;/
    <U00010837>..<U00010838>;<U0001083C>;<U0001083F>;/
    <U00010900>..<U00010919>;<U0001091F>;<U00010A00>..<U00010A03>;/
    <U00010A05>..<U00010A06>;<U00010A0C>..<U00010A13>;/
    <U00010A15>..<U00010A17>;<U00010A19>..<U00010A33>;/
    <U00010A38>..<U00010A3A>;<U00010A3F>..<U00010A47>;/
    <U00010A50>..<U00010A58>;<U00012000>..<U0001236E>;/
    <U00012400>..<U00012462>;<U00012470>..<U00012473>;/
    <U0001D000>..<U0001D0F5>;<U0001D100>..<U0001D126>;/
    <U0001D129>..<U0001D1DD>;<U0001D200>..<U0001D245>;/
    <U0001D300>..<U0001D356>;<U0001D360>..<U0001D371>;/
    <U0001D400>..<U0001D454>;<U0001D456>..<U0001D49C>;/
    <U0001D49E>..<U0001D49F>;<U0001D4A2>;<U0001D4A5>..<U0001D4A6>;/
    <U0001D4A9>..<U0001D4AC>;<U0001D4AE>..<U0001D4B9>;<U0001D4BB>;/
    <U0001D4BD>..<U0001D4C3>;<U0001D4C5>..<U0001D505>;/
    <U0001D507>..<U0001D50A>;<U0001D50D>..<U0001D514>;/
    <U0001D516>..<U0001D51C>;<U0001D51E>..<U0001D539>;/
    <U0001D53B>..<U0001D53E>;<U0001D540>..<U0001D544>;<U0001D546>;/
    <U0001D54A>..<U0001D550>;<U0001D552>..<U0001D6A5>;/
    <U0001D6A8>..<U0001D7CB>;<U0001D7CE>..<U0001D7FF>;/
    <U0001F000>..<U0001F02B>;<U0001F030>..<U0001F093>;/
    <U00020000>..<U0002A6D6>;<U0002F800>..<U0002FA1D>;<U000E0001>;/
    <U000E0020>..<U000E007F>;<U000E0100>..<U000E01EF>;/
    <U000F0000>..<U000FFFFD>;<U00100000>..<U0010FFFD>

print /
    <U0020>..<U007E>;<U00A0>..<U0377>;<U037A>..<U037E>;/
    <U0384>..<U038A>;<U038C>;<U038E>..<U03A1>;<U03A3>..<U0486>;/
    <U0488>..<U0523>;<U0531>..<U0556>;<U0559>..<U055F>;/
    <U0561>..<U0587>;<U0589>..<U058A>;<U0591>..<U05C7>;<U05D0>..<U05EA>;/
    <U05F0>..<U05F4>;<U0600>..<U0603>;<U0606>..<U061B>;/
    <U061E>..<U061F>;<U0621>..<U065E>;<U0660>..<U070D>;/
    <U070F>..<U074A>;<U074D>..<U07B1>;<U07C0>..<U07FA>;/
    <U0901>..<U0939>;<U093C>..<U094D>;<U0950>..<U0954>;<U0958>..<U0972>;/
    <U097B>..<U097F>;<U0981>..<U0983>;<U0985>..<U098C>;<U098F>..<U0990>;/
    <U0993>..<U09A8>;<U09AA>..<U09B0>;<U09B2>;<U09B6>..<U09B9>;/
    <U09BC>..<U09C4>;<U09C7>..<U09C8>;<U09CB>..<U09CE>;<U09D7>;/
    <U09DC>..<U09DD>;<U09DF>..<U09E3>;<U09E6>..<U09FA>;<U0A01>..<U0A03>;/
    <U0A05>..<U0A0A>;<U0A0F>..<U0A10>;<U0A13>..<U0A28>;<U0A2A>..<U0A30>;/
    <U0A32>..<U0A33>;<U0A35>..<U0A36>;<U0A38>..<U0A39>;<U0A3C>;/
    <U0A3E>..<U0A42>;<U0A47>..<U0A48>;<U0A4B>..<U0A4D>;<U0A51>;/
    <U0A59>..<U0A5C>;/
    <U0A5E>;<U0A66>..<U0A75>;<U0A81>..<U0A83>;<U0A85>..<U0A8D>;/
    <U0A8F>..<U0A91>;<U0A93>..<U0AA8>;<U0AAA>..<U0AB0>;<U0AB2>..<U0AB3>;/
    <U0AB5>..<U0AB9>;<U0ABC>..<U0AC5>;<U0AC7>..<U0AC9>;<U0ACB>..<U0ACD>;/
    <U0AD0>;<U0AE0>..<U0AE3>;<U0AE6>..<U0AEF>;<U0AF1>;<U0B01>..<U0B03>;/
    <U0B05>..<U0B0C>;<U0B0F>..<U0B10>;<U0B13>..<U0B28>;<U0B2A>..<U0B30>;/
    <U0B32>..<U0B33>;<U0B35>..<U0B39>;<U0B3C>..<U0B44>;<U0B47>..<U0B48>;/
    <U0B4B>..<U0B4D>;<U0B56>..<U0B57>;<U0B5C>..<U0B5D>;<U0B5F>..<U0B63>;/
    <U0B66>..<U0B71>;<U0B82>..<U0B83>;<U0B85>..<U0B8A>;<U0B8E>..<U0B90>;/
    <U0B92>..<U0B95>;<U0B99>..<U0B9A>;<U0B9C>;<U0B9E>..<U0B9F>;/
    <U0BA3>..<U0BA4>;<U0BA8>..<U0BAA>;<U0BAE>..<U0BB9>;<U0BBE>..<U0BC2>;/
```


<U0BC6>..<U0BC8>;<U0BCA>..<U0BCD>;<U0BD0>;<U0BD7>;<U0BE6>..<U0BFA>;/
<U0C01>..<U0C03>;<U0C05>..<U0C0C>;<U0C0E>..<U0C10>;<U0C12>..<U0C28>;/
<U0C2A>..<U0C33>;<U0C35>..<U0C39>;<U0C3D>..<U0C44>;<U0C46>..<U0C48>;/
<U0C4A>..<U0C4D>;<U0C55>..<U0C56>;<U0C58>..<U0C59>;<U0C60>..<U0C63>;/
<U0C66>..<U0C6F>;<U0C78>..<U0C7F>;/
<U0C82>..<U0C83>;<U0C85>..<U0C8C>;<U0C8E>..<U0C90>;<U0C92>..<U0CA8>;/
<U0CAA>..<U0CB3>;<U0CB5>..<U0CB9>;<U0CBC>..<U0CC4>;<U0CC6>..<U0CC8>;/
<U0CCA>..<U0CCD>;<U0CD5>..<U0CD6>;<U0CDE>;<U0CE0>..<U0CE3>;/
<U0CE6>..<U0CEF>;<U0CF1>..<U0CF2>;<U0D02>..<U0D03>;<U0D05>..<U0D0C>;/
<U0D0E>..<U0D10>;<U0D12>..<U0D28>;<U0D2A>..<U0D39>;<U0D3D>..<U0D44>;/
<U0D46>..<U0D48>;<U0D4A>..<U0D4D>;<U0D57>;<U0D60>..<U0D63>;/
<U0D66>..<U0D75>;<U0D79>..<U0D7F>;<U0D82>..<U0D83>;<U0D85>..<U0D96>;/
<U0D9A>..<U0DB1>;/
<U0DB3>..<U0DBB>;<U0DBD>;<U0DC0>..<U0DC6>;<U0DCA>;<U0DCF>..<U0DD4>;/
<U0DD6>;<U0DD8>..<U0DDF>;<U0DF2>..<U0DF4>;<U0E01>..<U0E3A>;/
<U0E3F>..<U0E5B>;<U0E81>..<U0E82>;<U0E84>;<U0E87>..<U0E88>;<U0E8A>;/
<U0E8D>;<U0E94>..<U0E97>;<U0E99>..<U0E9F>;<U0EA1>..<U0EA3>;<U0EA5>;/
<U0EA7>;<U0EAA>..<U0EAB>;<U0EAD>..<U0EB9>;<U0EBB>..<U0EBD>;/
<U0EC0>..<U0EC4>;<U0EC6>;<U0EC8>..<U0ECD>;<U0ED0>..<U0ED9>;/
<U0EDC>..<U0EDD>;<U0F00>..<U0F47>;<U0F49>..<U0F6C>;<U0F71>..<U0F8B>;/
<U0F90>..<U0F97>;<U0F99>..<U0FBC>;<U0FBE>..<U0FCC>;<U0FCE>..<U0FD4>;/
<U1000>..<U102A>;<U102B>..<U103F>;<U1040>..<U1099>;<U109E>;<U109F>;/
<U10A0>..<U10C5>;<U10D0>..<U10FC>;/
<U1100>..<U1159>;<U115F>..<U11A2>;<U11A8>..<U11F9>;<U1200>..<U1248>;/
<U124A>..<U124D>;<U1250>..<U1256>;<U1258>;<U125A>..<U125D>;/
<U1260>..<U1288>;<U128A>..<U128D>;<U1290>..<U12B0>;<U12B2>..<U12B5>;/
<U12B8>..<U12BE>;<U12C0>;<U12C2>..<U12C5>;<U12C8>..<U12D6>;/
<U12D8>..<U1310>;<U1312>..<U1315>;<U1318>..<U135A>;<U135F>..<U137C>;/
<U1380>..<U1399>;<U13A0>..<U13F4>;<U1401>..<U1676>;<U1680>..<U169C>;/
<U16A0>..<U16F0>;<U1700>..<U170C>;<U170E>..<U1714>;<U1720>..<U1736>;/
<U1740>..<U1753>;<U1760>..<U176C>;<U176E>..<U1770>;<U1772>..<U1773>;/
<U1780>..<U17DD>;<U17E0>..<U17E9>;<U17F0>..<U17F9>;<U1800>..<U180E>;/
<U1810>..<U1819>;<U1820>..<U1877>;<U1880>..<U18AA>;<U1900>..<U191C>;/
<U1920>..<U192B>;<U1930>..<U193B>;<U1940>;<U1944>..<U196D>;/
<U1970>..<U1974>;<U1980>..<U19A9>;<U19B0>..<U19C9>;<U19D0>..<U19D9>;/
<U19DE>..<U1A1B>;<U1A1E>..<U1A1F>;<U1B00>..<U1B4B>;<U1B50>..<U1B7C>;/
<U1B80>..<U1BAA>;<U1BAE>..<U1BB9>;<U1C00>..<U1C37>;<U1C3B>..<U1C49>;/
<U1C4D>..<U1C7F>;/
<U1D00>..<U1DE6>;<U1DFE>..<U1E9F>;<U1EA0>..<U1EFF>;<U1F00>..<U1F15>;/
<U1F18>..<U1F1D>;<U1F20>..<U1F45>;<U1F48>..<U1F4D>;<U1F50>..<U1F57>;/
<U1F59>;<U1F5B>;<U1F5D>;<U1F5F>..<U1F7D>;<U1F80>..<U1FB4>;/
<U1FB6>..<U1FC4>;<U1FC6>..<U1FD3>;<U1FD6>..<U1FDB>;<U1FDD>..<U1FEF>;/
<U1FF2>..<U1FF4>;<U1FF6>..<U1FFE>;<U2000>..<U2027>;<U202A>..<U2064>;/
<U206A>..<U2071>;<U2074>..<U208E>;<U2090>..<U2094>;<U20A0>..<U20B5>;/
<U20D0>..<U20F0>;<U2100>..<U214F>;<U2153>..<U2188>;<U2190>..<U23E7>;/
<U2400>..<U2426>;<U2440>..<U244A>;<U2460>..<U269D>;<U26A0>..<U26C3>;/
<U2701>..<U2704>;<U2706>..<U2709>;<U270C>..<U2727>;<U2729>..<U274B>;/
<U274D>;<U274F>..<U2752>;<U2756>;<U2758>..<U275E>;<U2761>..<U2794>;/
<U2798>..<U27AF>;<U27B1>..<U27BE>;<U27C0>..<U27CA>;<U27CC>;/
<U27D0>..<U27EF>;/
<U27F0>..<U2B4C>;<U2B50>..<U2B54>;<U2C00>..<U2C2E>;<U2C30>..<U2C5E>;/
<U2C60>..<U2C6F>;<U2C71>..<U2C7D>;<U2C80>..<U2CEA>;<U2CF9>..<U2D25>;/
<U2D30>..<U2D65>;<U2D6F>;<U2D80>..<U2D96>;<U2DA0>..<U2DA6>;/
<U2DA8>..<U2DAE>;<U2DB0>..<U2DB6>;<U2DB8>..<U2DBE>;<U2DC0>..<U2DC6>;/
<U2DC8>..<U2DCE>;<U2DD0>..<U2DD6>;<U2DD8>..<U2DDE>;<U2DE0>..<U2E30>;/
<U2E80>..<U2E99>;<U2E9B>..<U2EF3>;<U2F00>..<U2FD5>;/
<U2FF0>..<U2FFB>;<U3000>..<U303F>;<U3041>..<U3096>;<U3099>..<U30FF>;/
<U3105>..<U312D>;<U3131>..<U318E>;<U3190>..<U31B7>;<U31C0>..<U31CF>;/
<U31F0>..<U321E>;<U3220>..<U3243>;<U3250>..<U32FE>;<U3300>..<U4DB5>;/
<U4DC0>..<U9FBB>;<UA000>..<UA48C>;<UA490>..<UA4C6>;<UA500>..<UA62B>;/
<UA640>..<UA65F>;<UA662>..<UA673>;<UA67C>..<UA697>;<UA700>..<UA78C>;/
<UA7FB>..<UA7FF>;<UA800>..<UA82B>;<UA840>..<UA877>;/
<UA880>..<UA8C4>;<UA8CE>..<UA8D9>;<UAC00>..<UD7A3>;/
<UA840>..<UA877>;<UA880>..<UA8C4>;<UA8CE>..<UA8D9>;<UA900>..<UA953>;/
<UA95F>;<UAA00>..<UAA36>;<UAA40>..<UAA4D>;<UAA50>..<UAA59>;/


```
   <UAA5C>..<UAA5F>;/
   <UE000>..<UFA2D>;<UFA30>..<UFA6A>;<UFA70>..<UFAD9>;<UFB00>..<UFB06>;/
   <UFB13>..<UFB17>;<UFB1D>..<UFB36>;<UFB38>..<UFB3C>;<UFB3E>;/
   <UFB40>..<UFB41>;<UFB43>..<UFB44>;<UFB46>..<UFBB1>;<UFBD3>..<UFD3F>;/
   <UFD50>..<UFD8F>;<UFD92>..<UFDC7>;<UFDF0>..<UFDFD>;<UFE00>..<UFE19>;/
   <UFE20>..<UFE26>;<UFE30>..<UFE52>;<UFE54>..<UFE66>;<UFE68>..<UFE6B>;/
   <UFE70>..<UFE74>;<UFE76>..<UFEFC>;<UFEFF>;<UFF01>..<UFFBE>;/
   <UFFC2>..<UFFC7>;<UFFCA>..<UFFCF>;<UFFD2>..<UFFD7>;<UFFDA>..<UFFDC>;/
   <UFFE0>..<UFFE6>;<UFFE8>..<UFFEE>;<UFFF9>..<UFFFD>;/
   <U00010000>..<U0001000B>;<U0001000D>..<U00010026>;/
   <U00010028>..<U0001003A>;<U0001003C>..<U0001003D>;/
   <U0001003F>..<U0001004D>;<U00010050>..<U0001005D>;/
   <U00010080>..<U000100FA>;<U00010100>..<U00010102>;/
   <U00010107>..<U00010133>;<U00010137>..<U0001018A>;/
   <U00010190>..<U0001019B>;<U000101D0>..<U000101FD>;/
   <U00010280>..<U0001029C>;<U000102A0>..<U000102D0>;/
   <U00010300>..<U0001031E>;<U00010320>..<U00010323>;/
   <U00010330>..<U0001034A>;<U00010380>..<U0001039D>;/
   <U0001039F>..<U000103C3>;<U000103C8>..<U000103D5>;/
   <U00010400>..<U0001049D>;<U000104A0>..<U000104A9>;/
   <U00010800>..<U00010805>;<U00010808>;<U0001080A>..<U00010835>;/
   <U00010837>..<U00010838>;<U0001083C>;<U0001083F>;/
   <U00010900>..<U00010919>;<U0001091F>;<U00010A00>..<U00010A03>;/
   <U00010A05>..<U00010A06>;<U00010A0C>..<U00010A13>;/
   <U00010A15>..<U00010A17>;<U00010A19>..<U00010A33>;/
   <U00010A38>..<U00010A3A>;<U00010A3F>..<U00010A47>;/
   <U00010A50>..<U00010A58>;<U00012000>..<U0001236E>;/
   <U00012400>..<U00012462>;<U00012470>..<U00012473>;/
   <U0001D000>..<U0001D0F5>;<U0001D100>..<U0001D126>;/
   <U0001D129>..<U0001D1DD>;<U0001D200>..<U0001D245>;/
   <U0001D300>..<U0001D356>;<U0001D360>..<U0001D371>;/
   <U0001D400>..<U0001D454>;<U0001D456>..<U0001D49C>;/
   <U0001D49E>..<U0001D49F>;<U0001D4A2>;<U0001D4A5>..<U0001D4A6>;/
   <U0001D4A9>..<U0001D4AC>;<U0001D4AE>..<U0001D4B9>;<U0001D4BB>;/
   <U0001D4BD>..<U0001D4C3>;<U0001D4C5>..<U0001D505>;/
   <U0001D507>..<U0001D50A>;<U0001D50D>..<U0001D514>;/
   <U0001D516>..<U0001D51C>;<U0001D51E>..<U0001D539>;/
   <U0001D53B>..<U0001D53E>;<U0001D540>..<U0001D544>;<U0001D546>;/
   <U0001D54A>..<U0001D550>;<U0001D552>..<U0001D6A5>;/
   <U0001D6A8>..<U0001D7CB>;<U0001D7CE>..<U0001D7FF>;/
   <U0001F000>..<U0001F02B>;<U0001F030>..<U0001F093>;/
   <U00020000>..<U0002A6D6>;<U0002F800>..<U0002FA1D>;<U000E0001>;/
   <U000E0020>..<U000E007F>;<U000E0100>..<U000E01EF>;/
   <U000F0000>..<U000FFFFD>;<U00100000>..<U0010FFFD>

% The "xdigit" class must only contain the BASIC LATIN digits and A-F, a-f,
% says ISO C 99 (sections 7.25.2.1.12 and 6.4.4.1).
xdigit /
   <U0030>..<U0039>;<U0041>..<U0046>;<U0061>..<U0066>

blank /
   <U0009>;<U0020>;<U1680>;<U180E>;<U2000>..<U2006>;<U2008>..<U200A>;/
   <U205F>;<U3000>

toupper /
   (<U0061>,<U0041>);(<U0062>,<U0042>);(<U0063>,<U0043>);(<U0064>,<U0044>);/
   (<U0065>,<U0045>);(<U0066>,<U0046>);(<U0067>,<U0047>);(<U0068>,<U0048>);/
   (<U0069>,<U0049>);(<U006A>,<U004A>);(<U006B>,<U004B>);(<U006C>,<U004C>);/
   (<U006D>,<U004D>);(<U006E>,<U004E>);(<U006F>,<U004F>);(<U0070>,<U0050>);/
   (<U0071>,<U0051>);(<U0072>,<U0052>);(<U0073>,<U0053>);(<U0074>,<U0054>);/
   (<U0075>,<U0055>);(<U0076>,<U0056>);(<U0077>,<U0057>);(<U0078>,<U0058>);/
   (<U0079>,<U0059>);(<U007A>,<U005A>);(<U00B5>,<U039C>);(<U00E0>,<U00C0>);/
   (<U00E1>,<U00C1>);(<U00E2>,<U00C2>);(<U00E3>,<U00C3>);(<U00E4>,<U00C4>);/
   (<U00E5>,<U00C5>);(<U00E6>,<U00C6>);(<U00E7>,<U00C7>);(<U00E8>,<U00C8>);/
   (<U00E9>,<U00C9>);(<U00EA>,<U00CA>);(<U00EB>,<U00CB>);(<U00EC>,<U00CC>);/
```



```
(<U00ED>,<U00CD>);(<U00EE>,<U00CE>);(<U00EF>,<U00CF>);(<U00F0>,<U00D0>);/
(<U00F1>,<U00D1>);(<U00F2>,<U00D2>);(<U00F3>,<U00D3>);(<U00F4>,<U00D4>);/
(<U00F5>,<U00D5>);(<U00F6>,<U00D6>);(<U00F8>,<U00D8>);(<U00F9>,<U00D9>);/
(<U00FA>,<U00DA>);(<U00FB>,<U00DB>);(<U00FC>,<U00DC>);(<U00FD>,<U00DD>);/
(<U00FE>,<U00DE>);(<U00FF>,<U0178>);(<U0101>,<U0100>);(<U0103>,<U0102>);/
(<U0105>,<U0104>);(<U0107>,<U0106>);(<U0109>,<U0108>);(<U010B>,<U010A>);/
(<U010D>,<U010C>);(<U010F>,<U010E>);(<U0111>,<U0110>);(<U0113>,<U0112>);/
(<U0115>,<U0114>);(<U0117>,<U0116>);(<U0119>,<U0118>);(<U011B>,<U011A>);/
(<U011D>,<U011C>);(<U011F>,<U011E>);(<U0121>,<U0120>);(<U0123>,<U0122>);/
(<U0125>,<U0124>);(<U0127>,<U0126>);(<U0129>,<U0128>);(<U012B>,<U012A>);/
(<U012D>,<U012C>);(<U012F>,<U012E>);(<U0131>,<U0049>);(<U0133>,<U0132>);/
(<U0135>,<U0134>);(<U0137>,<U0136>);(<U013A>,<U0139>);(<U013C>,<U013B>);/
(<U013E>,<U013D>);(<U0140>,<U013F>);(<U0142>,<U0141>);(<U0144>,<U0143>);/
(<U0146>,<U0145>);(<U0148>,<U0147>);(<U014B>,<U014A>);(<U014D>,<U014C>);/
(<U014F>,<U014E>);(<U0151>,<U0150>);(<U0153>,<U0152>);(<U0155>,<U0154>);/
(<U0157>,<U0156>);(<U0159>,<U0158>);(<U015B>,<U015A>);(<U015D>,<U015C>);/
(<U015F>,<U015E>);(<U0161>,<U0160>);(<U0163>,<U0162>);(<U0165>,<U0164>);/
(<U0167>,<U0166>);(<U0169>,<U0168>);(<U016B>,<U016A>);(<U016D>,<U016C>);/
(<U016F>,<U016E>);(<U0171>,<U0170>);(<U0173>,<U0172>);(<U0175>,<U0174>);/
(<U0177>,<U0176>);(<U017A>,<U0179>);(<U017C>,<U017B>);(<U017E>,<U017D>);/
(<U017F>,<U0053>);(<U0180>,<U0243>);(<U0183>,<U0182>);(<U0185>,<U0184>);/
(<U0188>,<U0187>);(<U018C>,<U018B>);(<U0192>,<U0191>);(<U0195>,<U01F6>);/
(<U0199>,<U0198>);(<U019A>,<U023D>);(<U019E>,<U0220>);(<U01A1>,<U01A0>);/
(<U01A3>,<U01A2>);(<U01A5>,<U01A4>);(<U01A8>,<U01A7>);(<U01AD>,<U01AC>);/
(<U01B0>,<U01AF>);(<U01B4>,<U01B3>);(<U01B6>,<U01B5>);(<U01B9>,<U01B8>);/
(<U01BD>,<U01BC>);(<U01BF>,<U01F7>);(<U01C5>,<U01C4>);(<U01C6>,<U01C4>);/
(<U01C8>,<U01C7>);(<U01C9>,<U01C7>);(<U01CB>,<U01CA>);(<U01CC>,<U01CA>);/
(<U01CE>,<U01CD>);(<U01D0>,<U01CF>);(<U01D2>,<U01D1>);(<U01D4>,<U01D3>);/
(<U01D6>,<U01D5>);(<U01D8>,<U01D7>);(<U01DA>,<U01D9>);(<U01DC>,<U01DB>);/
(<U01DD>,<U018E>);(<U01DF>,<U01DE>);(<U01E1>,<U01E0>);(<U01E3>,<U01E2>);/
(<U01E5>,<U01E4>);(<U01E7>,<U01E6>);(<U01E9>,<U01E8>);(<U01EB>,<U01EA>);/
(<U01ED>,<U01EC>);(<U01EF>,<U01EE>);(<U01F2>,<U01F1>);(<U01F3>,<U01F1>);/
(<U01F5>,<U01F4>);(<U01F9>,<U01F8>);(<U01FB>,<U01FA>);(<U01FD>,<U01FC>);/
(<U01FF>,<U01FE>);(<U0201>,<U0200>);(<U0203>,<U0202>);(<U0205>,<U0204>);/
(<U0207>,<U0206>);(<U0209>,<U0208>);(<U020B>,<U020A>);(<U020D>,<U020C>);/
(<U020F>,<U020E>);(<U0211>,<U0210>);(<U0213>,<U0212>);(<U0215>,<U0214>);/
(<U0217>,<U0216>);(<U0219>,<U0218>);(<U021B>,<U021A>);(<U021D>,<U021C>);/
(<U021F>,<U021E>);(<U0223>,<U0222>);(<U0225>,<U0224>);(<U0227>,<U0226>);/
(<U0229>,<U0228>);(<U022B>,<U022A>);(<U022D>,<U022C>);(<U022F>,<U022E>);/
(<U0231>,<U0230>);(<U0233>,<U0232>);(<U023C>,<U023B>);(<U0242>,<U0241>);/
(<U0247>,<U0246>);(<U0249>,<U0248>);(<U024B>,<U024A>);(<U024D>,<U024C>);/
(<U024F>,<U024E>);(<U0250>,<U2C6F>);(<U0251>,<U2C6D>);(<U0253>,<U0181>);/
(<U0254>,<U0186>);(<U0256>,<U0189>);/
(<U0257>,<U018A>);(<U0259>,<U018F>);(<U025B>,<U0190>);(<U0260>,<U0193>);/
(<U0263>,<U0194>);(<U0268>,<U0197>);(<U0269>,<U0196>);(<U026B>,<U2C62>);/
(<U026F>,<U019C>);(<U0271>,<U2C6E>);(<U0272>,<U019D>);(<U0275>,<U019F>);/
(<U027D>,<U2C64>);/
(<U0280>,<U01A6>);(<U0283>,<U01A9>);(<U0288>,<U01AE>);(<U0289>,<U0244>);/
(<U028A>,<U01B1>);(<U028B>,<U01B2>);(<U028C>,<U0245>);(<U0292>,<U01B7>);/
(<U0345>,<U0399>);(<U0371>,<U0370>);(<U0373>,<U0372>);(<U0377>,<U0376>);/
(<U037B>,<U03FD>);(<U037C>,<U03FE>);(<U037D>,<U03FF>);/
(<U03AC>,<U0386>);(<U03AD>,<U0388>);(<U03AE>,<U0389>);(<U03AF>,<U038A>);/
(<U03B1>,<U0391>);(<U03B2>,<U0392>);(<U03B3>,<U0393>);(<U03B4>,<U0394>);/
(<U03B5>,<U0395>);(<U03B6>,<U0396>);(<U03B7>,<U0397>);(<U03B8>,<U0398>);/
(<U03B9>,<U0399>);(<U03BA>,<U039A>);(<U03BB>,<U039B>);(<U03BC>,<U039C>);/
(<U03BD>,<U039D>);(<U03BE>,<U039E>);(<U03BF>,<U039F>);(<U03C0>,<U03A0>);/
(<U03C1>,<U03A1>);(<U03C2>,<U03A3>);(<U03C3>,<U03A3>);(<U03C4>,<U03A4>);/
(<U03C5>,<U03A5>);(<U03C6>,<U03A6>);(<U03C7>,<U03A7>);(<U03C8>,<U03A8>);/
(<U03C9>,<U03A9>);(<U03CA>,<U03AA>);(<U03CB>,<U03AB>);(<U03CC>,<U038C>);/
(<U03CD>,<U038E>);(<U03CE>,<U038F>);(<U03D0>,<U0392>);(<U03D1>,<U0398>);/
(<U03D5>,<U03A6>);(<U03D6>,<U03A0>);(<U03D9>,<U03D8>);(<U03DB>,<U03DA>);/
(<U03DD>,<U03DC>);(<U03DF>,<U03DE>);(<U03E1>,<U03E0>);(<U03E3>,<U03E2>);/
(<U03E5>,<U03E4>);(<U03E7>,<U03E6>);(<U03E9>,<U03E8>);(<U03EB>,<U03EA>);/
(<U03ED>,<U03EC>);(<U03EF>,<U03EE>);(<U03F0>,<U039A>);(<U03F1>,<U03A1>);/
(<U03F2>,<U03F9>);(<U03F5>,<U0395>);(<U03F8>,<U03F7>);(<U03FB>,<U03FA>);/
```



```
(<U0430>,<U0410>);(<U0431>,<U0411>);(<U0432>,<U0412>);(<U0433>,<U0413>);/
(<U0434>,<U0414>);(<U0435>,<U0415>);(<U0436>,<U0416>);(<U0437>,<U0417>);/
(<U0438>,<U0418>);(<U0439>,<U0419>);(<U043A>,<U041A>);(<U043B>,<U041B>);/
(<U043C>,<U041C>);(<U043D>,<U041D>);(<U043E>,<U041E>);(<U043F>,<U041F>);/
(<U0440>,<U0420>);(<U0441>,<U0421>);(<U0442>,<U0422>);(<U0443>,<U0423>);/
(<U0444>,<U0424>);(<U0445>,<U0425>);(<U0446>,<U0426>);(<U0447>,<U0427>);/
(<U0448>,<U0428>);(<U0449>,<U0429>);(<U044A>,<U042A>);(<U044B>,<U042B>);/
(<U044C>,<U042C>);(<U044D>,<U042D>);(<U044E>,<U042E>);(<U044F>,<U042F>);/
(<U0450>,<U0400>);(<U0451>,<U0401>);(<U0452>,<U0402>);(<U0453>,<U0403>);/
(<U0454>,<U0404>);(<U0455>,<U0405>);(<U0456>,<U0406>);(<U0457>,<U0407>);/
(<U0458>,<U0408>);(<U0459>,<U0409>);(<U045A>,<U040A>);(<U045B>,<U040B>);/
(<U045C>,<U040C>);(<U045D>,<U040D>);(<U045E>,<U040E>);(<U045F>,<U040F>);/
(<U0461>,<U0460>);(<U0463>,<U0462>);(<U0465>,<U0464>);(<U0467>,<U0466>);/
(<U0469>,<U0468>);(<U046B>,<U046A>);(<U046D>,<U046C>);(<U046F>,<U046E>);/
(<U0471>,<U0470>);(<U0473>,<U0472>);(<U0475>,<U0474>);(<U0477>,<U0476>);/
(<U0479>,<U0478>);(<U047B>,<U047A>);(<U047D>,<U047C>);(<U047F>,<U047E>);/
(<U0481>,<U0480>);(<U048B>,<U048A>);(<U048D>,<U048C>);(<U048F>,<U048E>);/
(<U0491>,<U0490>);(<U0493>,<U0492>);(<U0495>,<U0494>);(<U0497>,<U0496>);/
(<U0499>,<U0498>);(<U049B>,<U049A>);(<U049D>,<U049C>);(<U049F>,<U049E>);/
(<U04A1>,<U04A0>);(<U04A3>,<U04A2>);(<U04A5>,<U04A4>);(<U04A7>,<U04A6>);/
(<U04A9>,<U04A8>);(<U04AB>,<U04AA>);(<U04AD>,<U04AC>);(<U04AF>,<U04AE>);/
(<U04B1>,<U04B0>);(<U04B3>,<U04B2>);(<U04B5>,<U04B4>);(<U04B7>,<U04B6>);/
(<U04B9>,<U04B8>);(<U04BB>,<U04BA>);(<U04BD>,<U04BC>);(<U04BF>,<U04BE>);/
(<U04C2>,<U04C1>);(<U04C4>,<U04C3>);(<U04C6>,<U04C5>);(<U04C8>,<U04C7>);/
(<U04CA>,<U04C9>);(<U04CC>,<U04CB>);(<U04CE>,<U04CD>);(<U04CF>,<U04C0>);/
(<U04D1>,<U04D0>);(<U04D3>,<U04D2>);(<U04D5>,<U04D4>);(<U04D7>,<U04D6>);/
(<U04D9>,<U04D8>);(<U04DB>,<U04DA>);(<U04DD>,<U04DC>);(<U04DF>,<U04DE>);/
(<U04E1>,<U04E0>);(<U04E3>,<U04E2>);(<U04E5>,<U04E4>);(<U04E7>,<U04E6>);/
(<U04E9>,<U04E8>);(<U04EB>,<U04EA>);(<U04ED>,<U04EC>);(<U04EF>,<U04EE>);/
(<U04F1>,<U04F0>);(<U04F3>,<U04F2>);(<U04F5>,<U04F4>);(<U04F7>,<U04F6>);/
(<U04F9>,<U04F8>);(<U04FB>,<U04FA>);(<U04FD>,<U04FC>);(<U04FF>,<U04FE>);/
(<U0501>,<U0500>);(<U0503>,<U0502>);(<U0505>,<U0504>);(<U0507>,<U0506>);/
(<U0509>,<U0508>);(<U050B>,<U050A>);(<U050D>,<U050C>);(<U050F>,<U050E>);/
(<U0511>,<U0510>);(<U0513>,<U0512>);(<U0515>,<U0514>);(<U0517>,<U0516>);/
(<U0519>,<U0518>);(<U051B>,<U051A>);(<U051D>,<U051C>);(<U051F>,<U051E>);/
(<U0521>,<U0520>);(<U0523>,<U0522>);(<U0561>,<U0531>);(<U0562>,<U0532>);/
(<U0563>,<U0533>);(<U0564>,<U0534>);(<U0565>,<U0535>);(<U0566>,<U0536>);/
(<U0567>,<U0537>);(<U0568>,<U0538>);(<U0569>,<U0539>);(<U056A>,<U053A>);/
(<U056B>,<U053B>);(<U056C>,<U053C>);(<U056D>,<U053D>);(<U056E>,<U053E>);/
(<U056F>,<U053F>);(<U0570>,<U0540>);(<U0571>,<U0541>);(<U0572>,<U0542>);/
(<U0573>,<U0543>);(<U0574>,<U0544>);(<U0575>,<U0545>);(<U0576>,<U0546>);/
(<U0577>,<U0547>);(<U0578>,<U0548>);(<U0579>,<U0549>);(<U057A>,<U054A>);/
(<U057B>,<U054B>);(<U057C>,<U054C>);(<U057D>,<U054D>);(<U057E>,<U054E>);/
(<U057F>,<U054F>);(<U0580>,<U0550>);(<U0581>,<U0551>);(<U0582>,<U0552>);/
(<U0583>,<U0553>);(<U0584>,<U0554>);(<U0585>,<U0555>);(<U0586>,<U0556>);/
(<U1D7D>,<U2C63>);(<U1E01>,<U1E00>);(<U1E03>,<U1E02>);(<U1E05>,<U1E04>);/
(<U1E07>,<U1E06>);(<U1E09>,<U1E08>);(<U1E0B>,<U1E0A>);(<U1E0D>,<U1E0C>);/
(<U1E0F>,<U1E0E>);(<U1E11>,<U1E10>);(<U1E13>,<U1E12>);(<U1E15>,<U1E14>);/
(<U1E17>,<U1E16>);(<U1E19>,<U1E18>);(<U1E1B>,<U1E1A>);(<U1E1D>,<U1E1C>);/
(<U1E1F>,<U1E1E>);(<U1E21>,<U1E20>);(<U1E23>,<U1E22>);(<U1E25>,<U1E24>);/
(<U1E27>,<U1E26>);(<U1E29>,<U1E28>);(<U1E2B>,<U1E2A>);(<U1E2D>,<U1E2C>);/
(<U1E2F>,<U1E2E>);(<U1E31>,<U1E30>);(<U1E33>,<U1E32>);(<U1E35>,<U1E34>);/
(<U1E37>,<U1E36>);(<U1E39>,<U1E38>);(<U1E3B>,<U1E3A>);(<U1E3D>,<U1E3C>);/
(<U1E3F>,<U1E3E>);(<U1E41>,<U1E40>);(<U1E43>,<U1E42>);(<U1E45>,<U1E44>);/
(<U1E47>,<U1E46>);(<U1E49>,<U1E48>);(<U1E4B>,<U1E4A>);(<U1E4D>,<U1E4C>);/
(<U1E4F>,<U1E4E>);(<U1E51>,<U1E50>);(<U1E53>,<U1E52>);(<U1E55>,<U1E54>);/
(<U1E57>,<U1E56>);(<U1E59>,<U1E58>);(<U1E5B>,<U1E5A>);(<U1E5D>,<U1E5C>);/
(<U1E5F>,<U1E5E>);(<U1E61>,<U1E60>);(<U1E63>,<U1E62>);(<U1E65>,<U1E64>);/
(<U1E67>,<U1E66>);(<U1E69>,<U1E68>);(<U1E6B>,<U1E6A>);(<U1E6D>,<U1E6C>);/
(<U1E6F>,<U1E6E>);(<U1E71>,<U1E70>);(<U1E73>,<U1E72>);(<U1E75>,<U1E74>);/
(<U1E77>,<U1E76>);(<U1E79>,<U1E78>);(<U1E7B>,<U1E7A>);(<U1E7D>,<U1E7C>);/
(<U1E7F>,<U1E7E>);(<U1E81>,<U1E80>);(<U1E83>,<U1E82>);(<U1E85>,<U1E84>);/
(<U1E87>,<U1E86>);(<U1E89>,<U1E88>);(<U1E8B>,<U1E8A>);(<U1E8D>,<U1E8C>);/
(<U1E8F>,<U1E8E>);(<U1E91>,<U1E90>);(<U1E93>,<U1E92>);(<U1E95>,<U1E94>);/
(<U1E9B>,<U1E60>);(<U1EA1>,<U1EA0>);(<U1EA3>,<U1EA2>);(<U1EA5>,<U1EA4>);/
```



```
(<U1EA7>,<U1EA6>);(<U1EA9>,<U1EA8>);(<U1EAB>,<U1EAA>);(<U1EAD>,<U1EAC>);/
(<U1EAF>,<U1EAE>);(<U1EB1>,<U1EB0>);(<U1EB3>,<U1EB2>);(<U1EB5>,<U1EB4>);/
(<U1EB7>,<U1EB6>);(<U1EB9>,<U1EB8>);(<U1EBB>,<U1EBA>);(<U1EBD>,<U1EBC>);/
(<U1EBF>,<U1EBE>);(<U1EC1>,<U1EC0>);(<U1EC3>,<U1EC2>);(<U1EC5>,<U1EC4>);/
(<U1EC7>,<U1EC6>);(<U1EC9>,<U1EC8>);(<U1ECB>,<U1ECA>);(<U1ECD>,<U1ECC>);/
(<U1ECF>,<U1ECE>);(<U1ED1>,<U1ED0>);(<U1ED3>,<U1ED2>);(<U1ED5>,<U1ED4>);/
(<U1ED7>,<U1ED6>);(<U1ED9>,<U1ED8>);(<U1EDB>,<U1EDA>);(<U1EDD>,<U1EDC>);/
(<U1EDF>,<U1EDE>);(<U1EE1>,<U1EE0>);(<U1EE3>,<U1EE2>);(<U1EE5>,<U1EE4>);/
(<U1EE7>,<U1EE6>);(<U1EE9>,<U1EE8>);(<U1EEB>,<U1EEA>);(<U1EED>,<U1EEC>);/
(<U1EEF>,<U1EEE>);(<U1EF1>,<U1EF0>);(<U1EF3>,<U1EF2>);(<U1EF5>,<U1EF4>);/
(<U1EF7>,<U1EF6>);(<U1EF9>,<U1EF8>);(<U1EFB>,<U1EFA>);(<U1EFD>,<U1EFC>);/
(<U1EFF>,<U1EFE>);(<U1F00>,<U1F08>);(<U1F01>,<U1F09>);/
(<U1F02>,<U1F0A>);(<U1F03>,<U1F0B>);(<U1F04>,<U1F0C>);(<U1F05>,<U1F0D>);/
(<U1F06>,<U1F0E>);(<U1F07>,<U1F0F>);(<U1F10>,<U1F18>);(<U1F11>,<U1F19>);/
(<U1F12>,<U1F1A>);(<U1F13>,<U1F1B>);(<U1F14>,<U1F1C>);(<U1F15>,<U1F1D>);/
(<U1F20>,<U1F28>);(<U1F21>,<U1F29>);(<U1F22>,<U1F2A>);(<U1F23>,<U1F2B>);/
(<U1F24>,<U1F2C>);(<U1F25>,<U1F2D>);(<U1F26>,<U1F2E>);(<U1F27>,<U1F2F>);/
(<U1F30>,<U1F38>);(<U1F31>,<U1F39>);(<U1F32>,<U1F3A>);(<U1F33>,<U1F3B>);/
(<U1F34>,<U1F3C>);(<U1F35>,<U1F3D>);(<U1F36>,<U1F3E>);(<U1F37>,<U1F3F>);/
(<U1F40>,<U1F48>);(<U1F41>,<U1F49>);(<U1F42>,<U1F4A>);(<U1F43>,<U1F4B>);/
(<U1F44>,<U1F4C>);(<U1F45>,<U1F4D>);(<U1F51>,<U1F59>);(<U1F53>,<U1F5B>);/
(<U1F55>,<U1F5D>);(<U1F57>,<U1F5F>);(<U1F60>,<U1F68>);(<U1F61>,<U1F69>);/
(<U1F62>,<U1F6A>);(<U1F63>,<U1F6B>);(<U1F64>,<U1F6C>);(<U1F65>,<U1F6D>);/
(<U1F66>,<U1F6E>);(<U1F67>,<U1F6F>);(<U1F70>,<U1FBA>);(<U1F71>,<U1FBB>);/
(<U1F72>,<U1FC8>);(<U1F73>,<U1FC9>);(<U1F74>,<U1FCA>);(<U1F75>,<U1FCB>);/
(<U1F76>,<U1FDA>);(<U1F77>,<U1FDB>);(<U1F78>,<U1FF8>);(<U1F79>,<U1FF9>);/
(<U1F7A>,<U1FEA>);(<U1F7B>,<U1FEB>);(<U1F7C>,<U1FFA>);(<U1F7D>,<U1FFB>);/
(<U1F80>,<U1F88>);(<U1F81>,<U1F89>);(<U1F82>,<U1F8A>);(<U1F83>,<U1F8B>);/
(<U1F84>,<U1F8C>);(<U1F85>,<U1F8D>);(<U1F86>,<U1F8E>);(<U1F87>,<U1F8F>);/
(<U1F90>,<U1F98>);(<U1F91>,<U1F99>);(<U1F92>,<U1F9A>);(<U1F93>,<U1F9B>);/
(<U1F94>,<U1F9C>);(<U1F95>,<U1F9D>);(<U1F96>,<U1F9E>);(<U1F97>,<U1F9F>);/
(<U1FA0>,<U1FA8>);(<U1FA1>,<U1FA9>);(<U1FA2>,<U1FAA>);(<U1FA3>,<U1FAB>);/
(<U1FA4>,<U1FAC>);(<U1FA5>,<U1FAD>);(<U1FA6>,<U1FAE>);(<U1FA7>,<U1FAF>);/
(<U1FB0>,<U1FB8>);(<U1FB1>,<U1FB9>);(<U1FB3>,<U1FBC>);(<U1FBE>,<U0399>);/
(<U1FC3>,<U1FCC>);(<U1FD0>,<U1FD8>);(<U1FD1>,<U1FD9>);(<U1FE0>,<U1FE8>);/
(<U1FE1>,<U1FE9>);(<U1FE5>,<U1FEC>);(<U1FF3>,<U1FFC>);(<U214E>,<U2132>);/
(<U2170>,<U2160>);(<U2171>,<U2161>);(<U2172>,<U2162>);(<U2173>,<U2163>);/
(<U2174>,<U2164>);(<U2175>,<U2165>);(<U2176>,<U2166>);(<U2177>,<U2167>);/
(<U2178>,<U2168>);(<U2179>,<U2169>);(<U217A>,<U216A>);(<U217B>,<U216B>);/
(<U217C>,<U216C>);(<U217D>,<U216D>);(<U217E>,<U216E>);(<U217F>,<U216F>);/
(<U2184>,<U2183>);(<U24D0>,<U24B6>);(<U24D1>,<U24B7>);(<U24D2>,<U24B8>);/
(<U24D3>,<U24B9>);(<U24D4>,<U24BA>);(<U24D5>,<U24BB>);(<U24D6>,<U24BC>);/
(<U24D7>,<U24BD>);(<U24D8>,<U24BE>);(<U24D9>,<U24BF>);(<U24DA>,<U24C0>);/
(<U24DB>,<U24C1>);(<U24DC>,<U24C2>);(<U24DD>,<U24C3>);(<U24DE>,<U24C4>);/
(<U24DF>,<U24C5>);(<U24E0>,<U24C6>);(<U24E1>,<U24C7>);(<U24E2>,<U24C8>);/
(<U24E3>,<U24C9>);(<U24E4>,<U24CA>);(<U24E5>,<U24CB>);(<U24E6>,<U24CC>);/
(<U24E7>,<U24CD>);(<U24E8>,<U24CE>);(<U24E9>,<U24CF>);(<U2C30>,<U2C00>);/
(<U2C31>,<U2C01>);(<U2C32>,<U2C02>);(<U2C33>,<U2C03>);(<U2C34>,<U2C04>);/
(<U2C35>,<U2C05>);(<U2C36>,<U2C06>);(<U2C37>,<U2C07>);(<U2C38>,<U2C08>);/
(<U2C39>,<U2C09>);(<U2C3A>,<U2C0A>);(<U2C3B>,<U2C0B>);(<U2C3C>,<U2C0C>);/
(<U2C3D>,<U2C0D>);(<U2C3E>,<U2C0E>);(<U2C3F>,<U2C0F>);(<U2C40>,<U2C10>);/
(<U2C41>,<U2C11>);(<U2C42>,<U2C12>);(<U2C43>,<U2C13>);(<U2C44>,<U2C14>);/
(<U2C45>,<U2C15>);(<U2C46>,<U2C16>);(<U2C47>,<U2C17>);(<U2C48>,<U2C18>);/
(<U2C49>,<U2C19>);(<U2C4A>,<U2C1A>);(<U2C4B>,<U2C1B>);(<U2C4C>,<U2C1C>);/
(<U2C4D>,<U2C1D>);(<U2C4E>,<U2C1E>);(<U2C4F>,<U2C1F>);(<U2C50>,<U2C20>);/
(<U2C51>,<U2C21>);(<U2C52>,<U2C22>);(<U2C53>,<U2C23>);(<U2C54>,<U2C24>);/
(<U2C55>,<U2C25>);(<U2C56>,<U2C26>);(<U2C57>,<U2C27>);(<U2C58>,<U2C28>);/
(<U2C59>,<U2C29>);(<U2C5A>,<U2C2A>);(<U2C5B>,<U2C2B>);(<U2C5C>,<U2C2C>);/
(<U2C5D>,<U2C2D>);(<U2C5E>,<U2C2E>);(<U2C61>,<U2C60>);(<U2C65>,<U023A>);/
(<U2C66>,<U023E>);(<U2C68>,<U2C67>);(<U2C6A>,<U2C69>);(<U2C6C>,<U2C6B>);/
(<U2C73>,<U2C72>);/
(<U2C76>,<U2C75>);(<U2C81>,<U2C80>);(<U2C83>,<U2C82>);(<U2C85>,<U2C84>);/
(<U2C87>,<U2C86>);(<U2C89>,<U2C88>);(<U2C8B>,<U2C8A>);(<U2C8D>,<U2C8C>);/
(<U2C8F>,<U2C8E>);(<U2C91>,<U2C90>);(<U2C93>,<U2C92>);(<U2C95>,<U2C94>);/
(<U2C97>,<U2C96>);(<U2C99>,<U2C98>);(<U2C9B>,<U2C9A>);(<U2C9D>,<U2C9C>);/
```



```
   (<U2C9F>,<U2C9E>);(<U2CA1>,<U2CA0>);(<U2CA3>,<U2CA2>);(<U2CA5>,<U2CA4>);/
   (<U2CA7>,<U2CA6>);(<U2CA9>,<U2CA8>);(<U2CAB>,<U2CAA>);(<U2CAD>,<U2CAC>);/
   (<U2CAF>,<U2CAE>);(<U2CB1>,<U2CB0>);(<U2CB3>,<U2CB2>);(<U2CB5>,<U2CB4>);/
   (<U2CB7>,<U2CB6>);(<U2CB9>,<U2CB8>);(<U2CBB>,<U2CBA>);(<U2CBD>,<U2CBC>);/
   (<U2CBF>,<U2CBE>);(<U2CC1>,<U2CC0>);(<U2CC3>,<U2CC2>);(<U2CC5>,<U2CC4>);/
   (<U2CC7>,<U2CC6>);(<U2CC9>,<U2CC8>);(<U2CCB>,<U2CCA>);(<U2CCD>,<U2CCC>);/
   (<U2CCF>,<U2CCE>);(<U2CD1>,<U2CD0>);(<U2CD3>,<U2CD2>);(<U2CD5>,<U2CD4>);/
   (<U2CD7>,<U2CD6>);(<U2CD9>,<U2CD8>);(<U2CDB>,<U2CDA>);(<U2CDD>,<U2CDC>);/
   (<U2CDF>,<U2CDE>);(<U2CE1>,<U2CE0>);(<U2CE3>,<U2CE2>);(<U2D00>,<U10A0>);/
   (<U2D01>,<U10A1>);(<U2D02>,<U10A2>);(<U2D03>,<U10A3>);(<U2D04>,<U10A4>);/
   (<U2D05>,<U10A5>);(<U2D06>,<U10A6>);(<U2D07>,<U10A7>);(<U2D08>,<U10A8>);/
   (<U2D09>,<U10A9>);(<U2D0A>,<U10AA>);(<U2D0B>,<U10AB>);(<U2D0C>,<U10AC>);/
   (<U2D0D>,<U10AD>);(<U2D0E>,<U10AE>);(<U2D0F>,<U10AF>);(<U2D10>,<U10B0>);/
   (<U2D11>,<U10B1>);(<U2D12>,<U10B2>);(<U2D13>,<U10B3>);(<U2D14>,<U10B4>);/
   (<U2D15>,<U10B5>);(<U2D16>,<U10B6>);(<U2D17>,<U10B7>);(<U2D18>,<U10B8>);/
   (<U2D19>,<U10B9>);(<U2D1A>,<U10BA>);(<U2D1B>,<U10BB>);(<U2D1C>,<U10BC>);/
   (<U2D1D>,<U10BD>);(<U2D1E>,<U10BE>);(<U2D1F>,<U10BF>);(<U2D20>,<U10C0>);/
   (<U2D21>,<U10C1>);(<U2D22>,<U10C2>);(<U2D23>,<U10C3>);(<U2D24>,<U10C4>);/
   (<U2D25>,<U10C5>);(<UFF41>,<UFF21>);(<UFF42>,<UFF22>);(<UFF43>,<UFF23>);/
   (<UFF44>,<UFF24>);(<UFF45>,<UFF25>);(<UFF46>,<UFF26>);(<UFF47>,<UFF27>);/
   (<UFF48>,<UFF28>);(<UFF49>,<UFF29>);(<UFF4A>,<UFF2A>);(<UFF4B>,<UFF2B>);/
   (<UFF4C>,<UFF2C>);(<UFF4D>,<UFF2D>);(<UFF4E>,<UFF2E>);(<UFF4F>,<UFF2F>);/
   (<UFF50>,<UFF30>);(<UFF51>,<UFF31>);(<UFF52>,<UFF32>);(<UFF53>,<UFF33>);/
   (<UFF54>,<UFF34>);(<UFF55>,<UFF35>);(<UFF56>,<UFF36>);(<UFF57>,<UFF37>);/
   (<UFF58>,<UFF38>);(<UFF59>,<UFF39>);(<UFF5A>,<UFF3A>);/
   (<U00010428>,<U00010400>);(<U00010429>,<U00010401>);/
   (<U0001042A>,<U00010402>);(<U0001042B>,<U00010403>);/
   (<U0001042C>,<U00010404>);(<U0001042D>,<U00010405>);/
   (<U0001042E>,<U00010406>);(<U0001042F>,<U00010407>);/
   (<U00010430>,<U00010408>);(<U00010431>,<U00010409>);/
   (<U00010432>,<U0001040A>);(<U00010433>,<U0001040B>);/
   (<U00010434>,<U0001040C>);(<U00010435>,<U0001040D>);/
   (<U00010436>,<U0001040E>);(<U00010437>,<U0001040F>);/
   (<U00010438>,<U00010410>);(<U00010439>,<U00010411>);/
   (<U0001043A>,<U00010412>);(<U0001043B>,<U00010413>);/
   (<U0001043C>,<U00010414>);(<U0001043D>,<U00010415>);/
   (<U0001043E>,<U00010416>);(<U0001043F>,<U00010417>);/
   (<U00010440>,<U00010418>);(<U00010441>,<U00010419>);/
   (<U00010442>,<U0001041A>);(<U00010443>,<U0001041B>);/
   (<U00010444>,<U0001041C>);(<U00010445>,<U0001041D>);/
   (<U00010446>,<U0001041E>);(<U00010447>,<U0001041F>);/
   (<U00010448>,<U00010420>);(<U00010449>,<U00010421>);/
   (<U0001044A>,<U00010422>);(<U0001044B>,<U00010423>);/
   (<U0001044C>,<U00010424>);(<U0001044D>,<U00010425>);/
   (<U0001044E>,<U00010426>);(<U0001044F>,<U00010427>)

tolower /
   (<U0041>,<U0061>);(<U0042>,<U0062>);(<U0043>,<U0063>);(<U0044>,<U0064>);/
   (<U0045>,<U0065>);(<U0046>,<U0066>);(<U0047>,<U0067>);(<U0048>,<U0068>);/
   (<U0049>,<U0069>);(<U004A>,<U006A>);(<U004B>,<U006B>);(<U004C>,<U006C>);/
   (<U004D>,<U006D>);(<U004E>,<U006E>);(<U004F>,<U006F>);(<U0050>,<U0070>);/
   (<U0051>,<U0071>);(<U0052>,<U0072>);(<U0053>,<U0073>);(<U0054>,<U0074>);/
   (<U0055>,<U0075>);(<U0056>,<U0076>);(<U0057>,<U0077>);(<U0058>,<U0078>);/
   (<U0059>,<U0079>);(<U005A>,<U007A>);(<U00C0>,<U00E0>);(<U00C1>,<U00E1>);/
   (<U00C2>,<U00E2>);(<U00C3>,<U00E3>);(<U00C4>,<U00E4>);(<U00C5>,<U00E5>);/
   (<U00C6>,<U00E6>);(<U00C7>,<U00E7>);(<U00C8>,<U00E8>);(<U00C9>,<U00E9>);/
   (<U00CA>,<U00EA>);(<U00CB>,<U00EB>);(<U00CC>,<U00EC>);(<U00CD>,<U00ED>);/
   (<U00CE>,<U00EE>);(<U00CF>,<U00EF>);(<U00D0>,<U00F0>);(<U00D1>,<U00F1>);/
   (<U00D2>,<U00F2>);(<U00D3>,<U00F3>);(<U00D4>,<U00F4>);(<U00D5>,<U00F5>);/
   (<U00D6>,<U00F6>);(<U00D8>,<U00F8>);(<U00D9>,<U00F9>);(<U00DA>,<U00FA>);/
   (<U00DB>,<U00FB>);(<U00DC>,<U00FC>);(<U00DD>,<U00FD>);(<U00DE>,<U00FE>);/
   (<U0100>,<U0101>);(<U0102>,<U0103>);(<U0104>,<U0105>);(<U0106>,<U0107>);/
   (<U0108>,<U0109>);(<U010A>,<U010B>);(<U010C>,<U010D>);(<U010E>,<U010F>);/
   (<U0110>,<U0111>);(<U0112>,<U0113>);(<U0114>,<U0115>);(<U0116>,<U0117>);/
   (<U0118>,<U0119>);(<U011A>,<U011B>);(<U011C>,<U011D>);(<U011E>,<U011F>);/
```



```
(<U0120>,<U0121>);(<U0122>,<U0123>);(<U0124>,<U0125>);(<U0126>,<U0127>);/
(<U0128>,<U0129>);(<U012A>,<U012B>);(<U012C>,<U012D>);(<U012E>,<U012F>);/
(<U0130>,<U0069>);(<U0132>,<U0133>);(<U0134>,<U0135>);(<U0136>,<U0137>);/
(<U0139>,<U013A>);(<U013B>,<U013C>);(<U013D>,<U013E>);(<U013F>,<U0140>);/
(<U0141>,<U0142>);(<U0143>,<U0144>);(<U0145>,<U0146>);(<U0147>,<U0148>);/
(<U014A>,<U014B>);(<U014C>,<U014D>);(<U014E>,<U014F>);(<U0150>,<U0151>);/
(<U0152>,<U0153>);(<U0154>,<U0155>);(<U0156>,<U0157>);(<U0158>,<U0159>);/
(<U015A>,<U015B>);(<U015C>,<U015D>);(<U015E>,<U015F>);(<U0160>,<U0161>);/
(<U0162>,<U0163>);(<U0164>,<U0165>);(<U0166>,<U0167>);(<U0168>,<U0169>);/
(<U016A>,<U016B>);(<U016C>,<U016D>);(<U016E>,<U016F>);(<U0170>,<U0171>);/
(<U0172>,<U0173>);(<U0174>,<U0175>);(<U0176>,<U0177>);(<U0178>,<U00FF>);/
(<U0179>,<U017A>);(<U017B>,<U017C>);(<U017D>,<U017E>);(<U0181>,<U0253>);/
(<U0182>,<U0183>);(<U0184>,<U0185>);(<U0186>,<U0254>);(<U0187>,<U0188>);/
(<U0189>,<U0256>);(<U018A>,<U0257>);(<U018B>,<U018C>);(<U018E>,<U01DD>);/
(<U018F>,<U0259>);(<U0190>,<U025B>);(<U0191>,<U0192>);(<U0193>,<U0260>);/
(<U0194>,<U0263>);(<U0196>,<U0269>);(<U0197>,<U0268>);(<U0198>,<U0199>);/
(<U019C>,<U026F>);(<U019D>,<U0272>);(<U019F>,<U0275>);(<U01A0>,<U01A1>);/
(<U01A2>,<U01A3>);(<U01A4>,<U01A5>);(<U01A6>,<U0280>);(<U01A7>,<U01A8>);/
(<U01A9>,<U0283>);(<U01AC>,<U01AD>);(<U01AE>,<U0288>);(<U01AF>,<U01B0>);/
(<U01B1>,<U028A>);(<U01B2>,<U028B>);(<U01B3>,<U01B4>);(<U01B5>,<U01B6>);/
(<U01B7>,<U0292>);(<U01B8>,<U01B9>);(<U01BC>,<U01BD>);(<U01C4>,<U01C6>);/
(<U01C5>,<U01C6>);(<U01C7>,<U01C9>);(<U01C8>,<U01C9>);(<U01CA>,<U01CC>);/
(<U01CB>,<U01CC>);(<U01CD>,<U01CE>);(<U01CF>,<U01D0>);(<U01D1>,<U01D2>);/
(<U01D3>,<U01D4>);(<U01D5>,<U01D6>);(<U01D7>,<U01D8>);(<U01D9>,<U01DA>);/
(<U01DB>,<U01DC>);(<U01DE>,<U01DF>);(<U01E0>,<U01E1>);(<U01E2>,<U01E3>);/
(<U01E4>,<U01E5>);(<U01E6>,<U01E7>);(<U01E8>,<U01E9>);(<U01EA>,<U01EB>);/
(<U01EC>,<U01ED>);(<U01EE>,<U01EF>);(<U01F1>,<U01F3>);(<U01F2>,<U01F3>);/
(<U01F4>,<U01F5>);(<U01F6>,<U0195>);(<U01F7>,<U01BF>);(<U01F8>,<U01F9>);/
(<U01FA>,<U01FB>);(<U01FC>,<U01FD>);(<U01FE>,<U01FF>);(<U0200>,<U0201>);/
(<U0202>,<U0203>);(<U0204>,<U0205>);(<U0206>,<U0207>);(<U0208>,<U0209>);/
(<U020A>,<U020B>);(<U020C>,<U020D>);(<U020E>,<U020F>);(<U0210>,<U0211>);/
(<U0212>,<U0213>);(<U0214>,<U0215>);(<U0216>,<U0217>);(<U0218>,<U0219>);/
(<U021A>,<U021B>);(<U021C>,<U021D>);(<U021E>,<U021F>);(<U0220>,<U019E>);/
(<U0222>,<U0223>);(<U0224>,<U0225>);(<U0226>,<U0227>);(<U0228>,<U0229>);/
(<U022A>,<U022B>);(<U022C>,<U022D>);(<U022E>,<U022F>);(<U0230>,<U0231>);/
(<U0232>,<U0233>);(<U023A>,<U2C65>);(<U023B>,<U023C>);(<U023D>,<U019A>);/
(<U023E>,<U2C66>);(<U0241>,<U0242>);(<U0243>,<U0180>);(<U0244>,<U0289>);/
(<U0245>,<U028C>);(<U0246>,<U0247>);(<U0248>,<U0249>);(<U024A>,<U024B>);/
(<U024C>,<U024D>);(<U024E>,<U024F>);(<U0370>,<U0371>);(<U0372>,<U0373>);/
(<U0376>,<U0377>);(<U0386>,<U03AC>);(<U0388>,<U03AD>);/
(<U0389>,<U03AE>);(<U038A>,<U03AF>);(<U038C>,<U03CC>);(<U038E>,<U03CD>);/
(<U038F>,<U03CE>);(<U0391>,<U03B1>);(<U0392>,<U03B2>);(<U0393>,<U03B3>);/
(<U0394>,<U03B4>);(<U0395>,<U03B5>);(<U0396>,<U03B6>);(<U0397>,<U03B7>);/
(<U0398>,<U03B8>);(<U0399>,<U03B9>);(<U039A>,<U03BA>);(<U039B>,<U03BB>);/
(<U039C>,<U03BC>);(<U039D>,<U03BD>);(<U039E>,<U03BE>);(<U039F>,<U03BF>);/
(<U03A0>,<U03C0>);(<U03A1>,<U03C1>);(<U03A3>,<U03C3>);(<U03A4>,<U03C4>);/
(<U03A5>,<U03C5>);(<U03A6>,<U03C6>);(<U03A7>,<U03C7>);(<U03A8>,<U03C8>);/
(<U03A9>,<U03C9>);(<U03AA>,<U03CA>);(<U03AB>,<U03CB>);(<U03D8>,<U03D9>);/
(<U03DA>,<U03DB>);(<U03DC>,<U03DD>);(<U03DE>,<U03DF>);(<U03E0>,<U03E1>);/
(<U03E2>,<U03E3>);(<U03E4>,<U03E5>);(<U03E6>,<U03E7>);(<U03E8>,<U03E9>);/
(<U03EA>,<U03EB>);(<U03EC>,<U03ED>);(<U03EE>,<U03EF>);(<U03F4>,<U03B8>);/
(<U03F7>,<U03F8>);(<U03F9>,<U03F2>);(<U03FA>,<U03FB>);(<U03FD>,<U037B>);/
(<U03FE>,<U037C>);(<U03FF>,<U037D>);(<U0400>,<U0450>);(<U0401>,<U0451>);/
(<U0402>,<U0452>);(<U0403>,<U0453>);(<U0404>,<U0454>);(<U0405>,<U0455>);/
(<U0406>,<U0456>);(<U0407>,<U0457>);(<U0408>,<U0458>);(<U0409>,<U0459>);/
(<U040A>,<U045A>);(<U040B>,<U045B>);(<U040C>,<U045C>);(<U040D>,<U045D>);/
(<U040E>,<U045E>);(<U040F>,<U045F>);(<U0410>,<U0430>);(<U0411>,<U0431>);/
(<U0412>,<U0432>);(<U0413>,<U0433>);(<U0414>,<U0434>);(<U0415>,<U0435>);/
(<U0416>,<U0436>);(<U0417>,<U0437>);(<U0418>,<U0438>);(<U0419>,<U0439>);/
(<U041A>,<U043A>);(<U041B>,<U043B>);(<U041C>,<U043C>);(<U041D>,<U043D>);/
(<U041E>,<U043E>);(<U041F>,<U043F>);(<U0420>,<U0440>);(<U0421>,<U0441>);/
(<U0422>,<U0442>);(<U0423>,<U0443>);(<U0424>,<U0444>);(<U0425>,<U0445>);/
(<U0426>,<U0446>);(<U0427>,<U0447>);(<U0428>,<U0448>);(<U0429>,<U0449>);/
(<U042A>,<U044A>);(<U042B>,<U044B>);(<U042C>,<U044C>);(<U042D>,<U044D>);/
(<U042E>,<U044E>);(<U042F>,<U044F>);(<U0460>,<U0461>);(<U0462>,<U0463>);/
```



```
(<U0464>,<U0465>);(<U0466>,<U0467>);(<U0468>,<U0469>);(<U046A>,<U046B>);/
(<U046C>,<U046D>);(<U046E>,<U046F>);(<U0470>,<U0471>);(<U0472>,<U0473>);/
(<U0474>,<U0475>);(<U0476>,<U0477>);(<U0478>,<U0479>);(<U047A>,<U047B>);/
(<U047C>,<U047D>);(<U047E>,<U047F>);(<U0480>,<U0481>);(<U048A>,<U048B>);/
(<U048C>,<U048D>);(<U048E>,<U048F>);(<U0490>,<U0491>);(<U0492>,<U0493>);/
(<U0494>,<U0495>);(<U0496>,<U0497>);(<U0498>,<U0499>);(<U049A>,<U049B>);/
(<U049C>,<U049D>);(<U049E>,<U049F>);(<U04A0>,<U04A1>);(<U04A2>,<U04A3>);/
(<U04A4>,<U04A5>);(<U04A6>,<U04A7>);(<U04A8>,<U04A9>);(<U04AA>,<U04AB>);/
(<U04AC>,<U04AD>);(<U04AE>,<U04AF>);(<U04B0>,<U04B1>);(<U04B2>,<U04B3>);/
(<U04B4>,<U04B5>);(<U04B6>,<U04B7>);(<U04B8>,<U04B9>);(<U04BA>,<U04BB>);/
(<U04BC>,<U04BD>);(<U04BE>,<U04BF>);(<U04C0>,<U04CF>);(<U04C1>,<U04C2>);/
(<U04C3>,<U04C4>);(<U04C5>,<U04C6>);(<U04C7>,<U04C8>);(<U04C9>,<U04CA>);/
(<U04CB>,<U04CC>);(<U04CD>,<U04CE>);(<U04D0>,<U04D1>);(<U04D2>,<U04D3>);/
(<U04D4>,<U04D5>);(<U04D6>,<U04D7>);(<U04D8>,<U04D9>);(<U04DA>,<U04DB>);/
(<U04DC>,<U04DD>);(<U04DE>,<U04DF>);(<U04E0>,<U04E1>);(<U04E2>,<U04E3>);/
(<U04E4>,<U04E5>);(<U04E6>,<U04E7>);(<U04E8>,<U04E9>);(<U04EA>,<U04EB>);/
(<U04EC>,<U04ED>);(<U04EE>,<U04EF>);(<U04F0>,<U04F1>);(<U04F2>,<U04F3>);/
(<U04F4>,<U04F5>);(<U04F6>,<U04F7>);(<U04F8>,<U04F9>);(<U04FA>,<U04FB>);/
(<U04FC>,<U04FD>);(<U04FE>,<U04FF>);(<U0500>,<U0501>);(<U0502>,<U0503>);/
(<U0504>,<U0505>);(<U0506>,<U0507>);(<U0508>,<U0509>);(<U050A>,<U050B>);/
(<U050C>,<U050D>);(<U050E>,<U050F>);(<U0510>,<U0511>);(<U0512>,<U0513>);/
(<U0514>,<U0515>);(<U0516>,<U0517>);(<U0518>,<U0519>);(<U051A>,<U051B>);/
(<U051C>,<U051D>);(<U051E>,<U051F>);(<U0520>,<U0521>);(<U0522>,<U0523>);/
(<U0531>,<U0561>);(<U0532>,<U0562>);(<U0533>,<U0563>);(<U0534>,<U0564>);/
(<U0535>,<U0565>);(<U0536>,<U0566>);(<U0537>,<U0567>);(<U0538>,<U0568>);/
(<U0539>,<U0569>);(<U053A>,<U056A>);(<U053B>,<U056B>);(<U053C>,<U056C>);/
(<U053D>,<U056D>);(<U053E>,<U056E>);(<U053F>,<U056F>);(<U0540>,<U0570>);/
(<U0541>,<U0571>);(<U0542>,<U0572>);(<U0543>,<U0573>);(<U0544>,<U0574>);/
(<U0545>,<U0575>);(<U0546>,<U0576>);(<U0547>,<U0577>);(<U0548>,<U0578>);/
(<U0549>,<U0579>);(<U054A>,<U057A>);(<U054B>,<U057B>);(<U054C>,<U057C>);/
(<U054D>,<U057D>);(<U054E>,<U057E>);(<U054F>,<U057F>);(<U0550>,<U0580>);/
(<U0551>,<U0581>);(<U0552>,<U0582>);(<U0553>,<U0583>);(<U0554>,<U0584>);/
(<U0555>,<U0585>);(<U0556>,<U0586>);(<U10A0>,<U2D00>);(<U10A1>,<U2D01>);/
(<U10A2>,<U2D02>);(<U10A3>,<U2D03>);(<U10A4>,<U2D04>);(<U10A5>,<U2D05>);/
(<U10A6>,<U2D06>);(<U10A7>,<U2D07>);(<U10A8>,<U2D08>);(<U10A9>,<U2D09>);/
(<U10AA>,<U2D0A>);(<U10AB>,<U2D0B>);(<U10AC>,<U2D0C>);(<U10AD>,<U2D0D>);/
(<U10AE>,<U2D0E>);(<U10AF>,<U2D0F>);(<U10B0>,<U2D10>);(<U10B1>,<U2D11>);/
(<U10B2>,<U2D12>);(<U10B3>,<U2D13>);(<U10B4>,<U2D14>);(<U10B5>,<U2D15>);/
(<U10B6>,<U2D16>);(<U10B7>,<U2D17>);(<U10B8>,<U2D18>);(<U10B9>,<U2D19>);/
(<U10BA>,<U2D1A>);(<U10BB>,<U2D1B>);(<U10BC>,<U2D1C>);(<U10BD>,<U2D1D>);/
(<U10BE>,<U2D1E>);(<U10BF>,<U2D1F>);(<U10C0>,<U2D20>);(<U10C1>,<U2D21>);/
(<U10C2>,<U2D22>);(<U10C3>,<U2D23>);(<U10C4>,<U2D24>);(<U10C5>,<U2D25>);/
(<U1E00>,<U1E01>);(<U1E02>,<U1E03>);(<U1E04>,<U1E05>);(<U1E06>,<U1E07>);/
(<U1E08>,<U1E09>);(<U1E0A>,<U1E0B>);(<U1E0C>,<U1E0D>);(<U1E0E>,<U1E0F>);/
(<U1E10>,<U1E11>);(<U1E12>,<U1E13>);(<U1E14>,<U1E15>);(<U1E16>,<U1E17>);/
(<U1E18>,<U1E19>);(<U1E1A>,<U1E1B>);(<U1E1C>,<U1E1D>);(<U1E1E>,<U1E1F>);/
(<U1E20>,<U1E21>);(<U1E22>,<U1E23>);(<U1E24>,<U1E25>);(<U1E26>,<U1E27>);/
(<U1E28>,<U1E29>);(<U1E2A>,<U1E2B>);(<U1E2C>,<U1E2D>);(<U1E2E>,<U1E2F>);/
(<U1E30>,<U1E31>);(<U1E32>,<U1E33>);(<U1E34>,<U1E35>);(<U1E36>,<U1E37>);/
(<U1E38>,<U1E39>);(<U1E3A>,<U1E3B>);(<U1E3C>,<U1E3D>);(<U1E3E>,<U1E3F>);/
(<U1E40>,<U1E41>);(<U1E42>,<U1E43>);(<U1E44>,<U1E45>);(<U1E46>,<U1E47>);/
(<U1E48>,<U1E49>);(<U1E4A>,<U1E4B>);(<U1E4C>,<U1E4D>);(<U1E4E>,<U1E4F>);/
(<U1E50>,<U1E51>);(<U1E52>,<U1E53>);(<U1E54>,<U1E55>);(<U1E56>,<U1E57>);/
(<U1E58>,<U1E59>);(<U1E5A>,<U1E5B>);(<U1E5C>,<U1E5D>);(<U1E5E>,<U1E5F>);/
(<U1E60>,<U1E61>);(<U1E62>,<U1E63>);(<U1E64>,<U1E65>);(<U1E66>,<U1E67>);/
(<U1E68>,<U1E69>);(<U1E6A>,<U1E6B>);(<U1E6C>,<U1E6D>);(<U1E6E>,<U1E6F>);/
(<U1E70>,<U1E71>);(<U1E72>,<U1E73>);(<U1E74>,<U1E75>);(<U1E76>,<U1E77>);/
(<U1E78>,<U1E79>);(<U1E7A>,<U1E7B>);(<U1E7C>,<U1E7D>);(<U1E7E>,<U1E7F>);/
(<U1E80>,<U1E81>);(<U1E82>,<U1E83>);(<U1E84>,<U1E85>);(<U1E86>,<U1E87>);/
(<U1E88>,<U1E89>);(<U1E8A>,<U1E8B>);(<U1E8C>,<U1E8D>);(<U1E8E>,<U1E8F>);/
(<U1E90>,<U1E91>);(<U1E92>,<U1E93>);(<U1E94>,<U1E95>);(<U1E9E>,<U00DF>);/
(<U1EA0>,<U1EA1>);/
(<U1EA2>,<U1EA3>);(<U1EA4>,<U1EA5>);(<U1EA6>,<U1EA7>);(<U1EA8>,<U1EA9>);/
(<U1EAA>,<U1EAB>);(<U1EAC>,<U1EAD>);(<U1EAE>,<U1EAF>);(<U1EB0>,<U1EB1>);/
(<U1EB2>,<U1EB3>);(<U1EB4>,<U1EB5>);(<U1EB6>,<U1EB7>);(<U1EB8>,<U1EB9>);/
```



```
(<U1EBA>,<U1EBB>);(<U1EBC>,<U1EBD>);(<U1EBE>,<U1EBF>);(<U1EC0>,<U1EC1>);/
(<U1EC2>,<U1EC3>);(<U1EC4>,<U1EC5>);(<U1EC6>,<U1EC7>);(<U1EC8>,<U1EC9>);/
(<U1ECA>,<U1ECB>);(<U1ECC>,<U1ECD>);(<U1ECE>,<U1ECF>);(<U1ED0>,<U1ED1>);/
(<U1ED2>,<U1ED3>);(<U1ED4>,<U1ED5>);(<U1ED6>,<U1ED7>);(<U1ED8>,<U1ED9>);/
(<U1EDA>,<U1EDB>);(<U1EDC>,<U1EDD>);(<U1EDE>,<U1EDF>);(<U1EE0>,<U1EE1>);/
(<U1EE2>,<U1EE3>);(<U1EE4>,<U1EE5>);(<U1EE6>,<U1EE7>);(<U1EE8>,<U1EE9>);/
(<U1EEA>,<U1EEB>);(<U1EEC>,<U1EED>);(<U1EEE>,<U1EEF>);(<U1EF0>,<U1EF1>);/
(<U1EF2>,<U1EF3>);(<U1EF4>,<U1EF5>);(<U1EF6>,<U1EF7>);(<U1EF8>,<U1EF9>);/
(<U1EFA>,<U1EFB>);(<U1EFC>,<U1EFD>);(<U1EFE>,<U1EFF>);/
(<U1F08>,<U1F00>);(<U1F09>,<U1F01>);(<U1F0A>,<U1F02>);(<U1F0B>,<U1F03>);/
(<U1F0C>,<U1F04>);(<U1F0D>,<U1F05>);(<U1F0E>,<U1F06>);(<U1F0F>,<U1F07>);/
(<U1F18>,<U1F10>);(<U1F19>,<U1F11>);(<U1F1A>,<U1F12>);(<U1F1B>,<U1F13>);/
(<U1F1C>,<U1F14>);(<U1F1D>,<U1F15>);(<U1F28>,<U1F20>);(<U1F29>,<U1F21>);/
(<U1F2A>,<U1F22>);(<U1F2B>,<U1F23>);(<U1F2C>,<U1F24>);(<U1F2D>,<U1F25>);/
(<U1F2E>,<U1F26>);(<U1F2F>,<U1F27>);(<U1F38>,<U1F30>);(<U1F39>,<U1F31>);/
(<U1F3A>,<U1F32>);(<U1F3B>,<U1F33>);(<U1F3C>,<U1F34>);(<U1F3D>,<U1F35>);/
(<U1F3E>,<U1F36>);(<U1F3F>,<U1F37>);(<U1F48>,<U1F40>);(<U1F49>,<U1F41>);/
(<U1F4A>,<U1F42>);(<U1F4B>,<U1F43>);(<U1F4C>,<U1F44>);(<U1F4D>,<U1F45>);/
(<U1F59>,<U1F51>);(<U1F5B>,<U1F53>);(<U1F5D>,<U1F55>);(<U1F5F>,<U1F57>);/
(<U1F68>,<U1F60>);(<U1F69>,<U1F61>);(<U1F6A>,<U1F62>);(<U1F6B>,<U1F63>);/
(<U1F6C>,<U1F64>);(<U1F6D>,<U1F65>);(<U1F6E>,<U1F66>);(<U1F6F>,<U1F67>);/
(<U1F88>,<U1F80>);(<U1F89>,<U1F81>);(<U1F8A>,<U1F82>);(<U1F8B>,<U1F83>);/
(<U1F8C>,<U1F84>);(<U1F8D>,<U1F85>);(<U1F8E>,<U1F86>);(<U1F8F>,<U1F87>);/
(<U1F98>,<U1F90>);(<U1F99>,<U1F91>);(<U1F9A>,<U1F92>);(<U1F9B>,<U1F93>);/
(<U1F9C>,<U1F94>);(<U1F9D>,<U1F95>);(<U1F9E>,<U1F96>);(<U1F9F>,<U1F97>);/
(<U1FA8>,<U1FA0>);(<U1FA9>,<U1FA1>);(<U1FAA>,<U1FA2>);(<U1FAB>,<U1FA3>);/
(<U1FAC>,<U1FA4>);(<U1FAD>,<U1FA5>);(<U1FAE>,<U1FA6>);(<U1FAF>,<U1FA7>);/
(<U1FB8>,<U1FB0>);(<U1FB9>,<U1FB1>);(<U1FBA>,<U1F70>);(<U1FBB>,<U1F71>);/
(<U1FBC>,<U1FB3>);(<U1FC8>,<U1F72>);(<U1FC9>,<U1F73>);(<U1FCA>,<U1F74>);/
(<U1FCB>,<U1F75>);(<U1FCC>,<U1FC3>);(<U1FD8>,<U1FD0>);(<U1FD9>,<U1FD1>);/
(<U1FDA>,<U1F76>);(<U1FDB>,<U1F77>);(<U1FE8>,<U1FE0>);(<U1FE9>,<U1FE1>);/
(<U1FEA>,<U1F7A>);(<U1FEB>,<U1F7B>);(<U1FEC>,<U1FE5>);(<U1FF8>,<U1F78>);/
(<U1FF9>,<U1F79>);(<U1FFA>,<U1F7C>);(<U1FFB>,<U1F7D>);(<U1FFC>,<U1FF3>);/
(<U2126>,<U03C9>);(<U212A>,<U006B>);(<U212B>,<U00E5>);(<U2132>,<U214E>);/
(<U2160>,<U2170>);(<U2161>,<U2171>);(<U2162>,<U2172>);(<U2163>,<U2173>);/
(<U2164>,<U2174>);(<U2165>,<U2175>);(<U2166>,<U2176>);(<U2167>,<U2177>);/
(<U2168>,<U2178>);(<U2169>,<U2179>);(<U216A>,<U217A>);(<U216B>,<U217B>);/
(<U216C>,<U217C>);(<U216D>,<U217D>);(<U216E>,<U217E>);(<U216F>,<U217F>);/
(<U2183>,<U2184>);(<U24B6>,<U24D0>);(<U24B7>,<U24D1>);(<U24B8>,<U24D2>);/
(<U24B9>,<U24D3>);(<U24BA>,<U24D4>);(<U24BB>,<U24D5>);(<U24BC>,<U24D6>);/
(<U24BD>,<U24D7>);(<U24BE>,<U24D8>);(<U24BF>,<U24D9>);(<U24C0>,<U24DA>);/
(<U24C1>,<U24DB>);(<U24C2>,<U24DC>);(<U24C3>,<U24DD>);(<U24C4>,<U24DE>);/
(<U24C5>,<U24DF>);(<U24C6>,<U24E0>);(<U24C7>,<U24E1>);(<U24C8>,<U24E2>);/
(<U24C9>,<U24E3>);(<U24CA>,<U24E4>);(<U24CB>,<U24E5>);(<U24CC>,<U24E6>);/
(<U24CD>,<U24E7>);(<U24CE>,<U24E8>);(<U24CF>,<U24E9>);(<U2C00>,<U2C30>);/
(<U2C01>,<U2C31>);(<U2C02>,<U2C32>);(<U2C03>,<U2C33>);(<U2C04>,<U2C34>);/
(<U2C05>,<U2C35>);(<U2C06>,<U2C36>);(<U2C07>,<U2C37>);(<U2C08>,<U2C38>);/
(<U2C09>,<U2C39>);(<U2C0A>,<U2C3A>);(<U2C0B>,<U2C3B>);(<U2C0C>,<U2C3C>);/
(<U2C0D>,<U2C3D>);(<U2C0E>,<U2C3E>);(<U2C0F>,<U2C3F>);(<U2C10>,<U2C40>);/
(<U2C11>,<U2C41>);(<U2C12>,<U2C42>);(<U2C13>,<U2C43>);(<U2C14>,<U2C44>);/
(<U2C15>,<U2C45>);(<U2C16>,<U2C46>);(<U2C17>,<U2C47>);(<U2C18>,<U2C48>);/
(<U2C19>,<U2C49>);(<U2C1A>,<U2C4A>);(<U2C1B>,<U2C4B>);(<U2C1C>,<U2C4C>);/
(<U2C1D>,<U2C4D>);(<U2C1E>,<U2C4E>);(<U2C1F>,<U2C4F>);(<U2C20>,<U2C50>);/
(<U2C21>,<U2C51>);(<U2C22>,<U2C52>);(<U2C23>,<U2C53>);(<U2C24>,<U2C54>);/
(<U2C25>,<U2C55>);(<U2C26>,<U2C56>);(<U2C27>,<U2C57>);(<U2C28>,<U2C58>);/
(<U2C29>,<U2C59>);(<U2C2A>,<U2C5A>);(<U2C2B>,<U2C5B>);(<U2C2C>,<U2C5C>);/
(<U2C2D>,<U2C5D>);(<U2C2E>,<U2C5E>);(<U2C60>,<U2C61>);(<U2C62>,<U026B>);/
(<U2C63>,<U1D7D>);(<U2C64>,<U027D>);(<U2C67>,<U2C68>);(<U2C69>,<U2C6A>);/
(<U2C6B>,<U2C6C>);(<U2C6D>,<U0251>);(<U2C6E>,<U0271>);(<U2C6F>,<U0250>);/
(<U2C72>,<U2C73>);(<U2C75>,<U2C76>);(<U2C80>,<U2C81>);(<U2C82>,<U2C83>);/
(<U2C84>,<U2C85>);(<U2C86>,<U2C87>);(<U2C88>,<U2C89>);(<U2C8A>,<U2C8B>);/
(<U2C8C>,<U2C8D>);(<U2C8E>,<U2C8F>);(<U2C90>,<U2C91>);(<U2C92>,<U2C93>);/
(<U2C94>,<U2C95>);(<U2C96>,<U2C97>);(<U2C98>,<U2C99>);(<U2C9A>,<U2C9B>);/
(<U2C9C>,<U2C9D>);(<U2C9E>,<U2C9F>);(<U2CA0>,<U2CA1>);(<U2CA2>,<U2CA3>);/
(<U2CA4>,<U2CA5>);(<U2CA6>,<U2CA7>);(<U2CA8>,<U2CA9>);(<U2CAA>,<U2CAB>);/
```



```
    (<U2CAC>,<U2CAD>);(<U2CAE>,<U2CAF>);(<U2CB0>,<U2CB1>);(<U2CB2>,<U2CB3>);/
    (<U2CB4>,<U2CB5>);(<U2CB6>,<U2CB7>);(<U2CB8>,<U2CB9>);(<U2CBA>,<U2CBB>);/
    (<U2CBC>,<U2CBD>);(<U2CBE>,<U2CBF>);(<U2CC0>,<U2CC1>);(<U2CC2>,<U2CC3>);/
    (<U2CC4>,<U2CC5>);(<U2CC6>,<U2CC7>);(<U2CC8>,<U2CC9>);(<U2CCA>,<U2CCB>);/
    (<U2CCC>,<U2CCD>);(<U2CCE>,<U2CCF>);(<U2CD0>,<U2CD1>);(<U2CD2>,<U2CD3>);/
    (<U2CD4>,<U2CD5>);(<U2CD6>,<U2CD7>);(<U2CD8>,<U2CD9>);(<U2CDA>,<U2CDB>);/
    (<U2CDC>,<U2CDD>);(<U2CDE>,<U2CDF>);(<U2CE0>,<U2CE1>);(<U2CE2>,<U2CE3>);/
    (<UFF21>,<UFF41>);(<UFF22>,<UFF42>);(<UFF23>,<UFF43>);(<UFF24>,<UFF44>);/
    (<UFF25>,<UFF45>);(<UFF26>,<UFF46>);(<UFF27>,<UFF47>);(<UFF28>,<UFF48>);/
    (<UFF29>,<UFF49>);(<UFF2A>,<UFF4A>);(<UFF2B>,<UFF4B>);(<UFF2C>,<UFF4C>);/
    (<UFF2D>,<UFF4D>);(<UFF2E>,<UFF4E>);(<UFF2F>,<UFF4F>);(<UFF30>,<UFF50>);/
    (<UFF31>,<UFF51>);(<UFF32>,<UFF52>);(<UFF33>,<UFF53>);(<UFF34>,<UFF54>);/
    (<UFF35>,<UFF55>);(<UFF36>,<UFF56>);(<UFF37>,<UFF57>);(<UFF38>,<UFF58>);/
    (<UFF39>,<UFF59>);(<UFF3A>,<UFF5A>);(<U00010400>,<U00010428>);/
    (<U00010401>,<U00010429>);(<U00010402>,<U0001042A>);/
    (<U00010403>,<U0001042B>);(<U00010404>,<U0001042C>);/
    (<U00010405>,<U0001042D>);(<U00010406>,<U0001042E>);/
    (<U00010407>,<U0001042F>);(<U00010408>,<U00010430>);/
    (<U00010409>,<U00010431>);(<U0001040A>,<U00010432>);/
    (<U0001040B>,<U00010433>);(<U0001040C>,<U00010434>);/
    (<U0001040D>,<U00010435>);(<U0001040E>,<U00010436>);/
    (<U0001040F>,<U00010437>);(<U00010410>,<U00010438>);/
    (<U00010411>,<U00010439>);(<U00010412>,<U0001043A>);/
    (<U00010413>,<U0001043B>);(<U00010414>,<U0001043C>);/
    (<U00010415>,<U0001043D>);(<U00010416>,<U0001043E>);/
    (<U00010417>,<U0001043F>);(<U00010418>,<U00010440>);/
    (<U00010419>,<U00010441>);(<U0001041A>,<U00010442>);/
    (<U0001041B>,<U00010443>);(<U0001041C>,<U00010444>);/
    (<U0001041D>,<U00010445>);(<U0001041E>,<U00010446>);/
    (<U0001041F>,<U00010447>);(<U00010420>,<U00010448>);/
    (<U00010421>,<U00010449>);(<U00010422>,<U0001044A>);/
    (<U00010423>,<U0001044B>);(<U00010424>,<U0001044C>);/
    (<U00010425>,<U0001044D>);(<U00010426>,<U0001044E>);/
    (<U00010427>,<U0001044F>)

map "totitle"; /
    (<U0061>,<U0041>);(<U0062>,<U0042>);(<U0063>,<U0043>);(<U0064>,<U0044>);/
    (<U0065>,<U0045>);(<U0066>,<U0046>);(<U0067>,<U0047>);(<U0068>,<U0048>);/
    (<U0069>,<U0049>);(<U006A>,<U004A>);(<U006B>,<U004B>);(<U006C>,<U004C>);/
    (<U006D>,<U004D>);(<U006E>,<U004E>);(<U006F>,<U004F>);(<U0070>,<U0050>);/
    (<U0071>,<U0051>);(<U0072>,<U0052>);(<U0073>,<U0053>);(<U0074>,<U0054>);/
    (<U0075>,<U0055>);(<U0076>,<U0056>);(<U0077>,<U0057>);(<U0078>,<U0058>);/
    (<U0079>,<U0059>);(<U007A>,<U005A>);(<U00B5>,<U039C>);(<U00E0>,<U00C0>);/
    (<U00E1>,<U00C1>);(<U00E2>,<U00C2>);(<U00E3>,<U00C3>);(<U00E4>,<U00C4>);/
    (<U00E5>,<U00C5>);(<U00E6>,<U00C6>);(<U00E7>,<U00C7>);(<U00E8>,<U00C8>);/
    (<U00E9>,<U00C9>);(<U00EA>,<U00CA>);(<U00EB>,<U00CB>);(<U00EC>,<U00CC>);/
    (<U00ED>,<U00CD>);(<U00EE>,<U00CE>);(<U00EF>,<U00CF>);(<U00F0>,<U00D0>);/
    (<U00F1>,<U00D1>);(<U00F2>,<U00D2>);(<U00F3>,<U00D3>);(<U00F4>,<U00D4>);/
    (<U00F5>,<U00D5>);(<U00F6>,<U00D6>);(<U00F8>,<U00D8>);(<U00F9>,<U00D9>);/
    (<U00FA>,<U00DA>);(<U00FB>,<U00DB>);(<U00FC>,<U00DC>);(<U00FD>,<U00DD>);/
    (<U00FE>,<U00DE>);(<U00FF>,<U0178>);(<U0101>,<U0100>);(<U0103>,<U0102>);/
    (<U0105>,<U0104>);(<U0107>,<U0106>);(<U0109>,<U0108>);(<U010B>,<U010A>);/
    (<U010D>,<U010C>);(<U010F>,<U010E>);(<U0111>,<U0110>);(<U0113>,<U0112>);/
    (<U0115>,<U0114>);(<U0117>,<U0116>);(<U0119>,<U0118>);(<U011B>,<U011A>);/
    (<U011D>,<U011C>);(<U011F>,<U011E>);(<U0121>,<U0120>);(<U0123>,<U0122>);/
    (<U0125>,<U0124>);(<U0127>,<U0126>);(<U0129>,<U0128>);(<U012B>,<U012A>);/
    (<U012D>,<U012C>);(<U012F>,<U012E>);(<U0131>,<U0049>);(<U0133>,<U0132>);/
    (<U0135>,<U0134>);(<U0137>,<U0136>);(<U013A>,<U0139>);(<U013C>,<U013B>);/
    (<U013E>,<U013D>);(<U0140>,<U013F>);(<U0142>,<U0141>);(<U0144>,<U0143>);/
    (<U0146>,<U0145>);(<U0148>,<U0147>);(<U014B>,<U014A>);(<U014D>,<U014C>);/
    (<U014F>,<U014E>);(<U0151>,<U0150>);(<U0153>,<U0152>);(<U0155>,<U0154>);/
    (<U0157>,<U0156>);(<U0159>,<U0158>);(<U015B>,<U015A>);(<U015D>,<U015C>);/
    (<U015F>,<U015E>);(<U0161>,<U0160>);(<U0163>,<U0162>);(<U0165>,<U0164>);/
    (<U0167>,<U0166>);(<U0169>,<U0168>);(<U016B>,<U016A>);(<U016D>,<U016C>);/
    (<U016F>,<U016E>);(<U0171>,<U0170>);(<U0173>,<U0172>);(<U0175>,<U0174>);/
```



```
(<U0177>,<U0176>);(<U017A>,<U0179>);(<U017C>,<U017B>);(<U017E>,<U017D>);/
(<U017F>,<U0053>);(<U0180>,<U0243>);(<U0183>,<U0182>);(<U0185>,<U0184>);/
(<U0188>,<U0187>);(<U018C>,<U018B>);(<U0192>,<U0191>);(<U0195>,<U01F6>);/
(<U0199>,<U0198>);(<U019A>,<U023D>);(<U019E>,<U0220>);(<U01A1>,<U01A0>);/
(<U01A3>,<U01A2>);(<U01A5>,<U01A4>);(<U01A8>,<U01A7>);(<U01AD>,<U01AC>);/
(<U01B0>,<U01AF>);(<U01B4>,<U01B3>);(<U01B6>,<U01B5>);(<U01B9>,<U01B8>);/
(<U01BD>,<U01BC>);(<U01BF>,<U01F7>);(<U01C4>,<U01C5>);(<U01C6>,<U01C5>);/
(<U01C7>,<U01C8>);(<U01C9>,<U01C8>);(<U01CA>,<U01CB>);(<U01CC>,<U01CB>);/
(<U01CE>,<U01CD>);(<U01D0>,<U01CF>);(<U01D2>,<U01D1>);(<U01D4>,<U01D3>);/
(<U01D6>,<U01D5>);(<U01D8>,<U01D7>);(<U01DA>,<U01D9>);(<U01DC>,<U01DB>);/
(<U01DD>,<U018E>);(<U01DF>,<U01DE>);(<U01E1>,<U01E0>);(<U01E3>,<U01E2>);/
(<U01E5>,<U01E4>);(<U01E7>,<U01E6>);(<U01E9>,<U01E8>);(<U01EB>,<U01EA>);/
(<U01ED>,<U01EC>);(<U01EF>,<U01EE>);(<U01F1>,<U01F2>);(<U01F3>,<U01F2>);/
(<U01F5>,<U01F4>);(<U01F9>,<U01F8>);(<U01FB>,<U01FA>);(<U01FD>,<U01FC>);/
(<U01FF>,<U01FE>);(<U0201>,<U0200>);(<U0203>,<U0202>);(<U0205>,<U0204>);/
(<U0207>,<U0206>);(<U0209>,<U0208>);(<U020B>,<U020A>);(<U020D>,<U020C>);/
(<U020F>,<U020E>);(<U0211>,<U0210>);(<U0213>,<U0212>);(<U0215>,<U0214>);/
(<U0217>,<U0216>);(<U0219>,<U0218>);(<U021B>,<U021A>);(<U021D>,<U021C>);/
(<U021F>,<U021E>);(<U0223>,<U0222>);(<U0225>,<U0224>);(<U0227>,<U0226>);/
(<U0229>,<U0228>);(<U022B>,<U022A>);(<U022D>,<U022C>);(<U022F>,<U022E>);/
(<U0231>,<U0230>);(<U0233>,<U0232>);(<U023C>,<U023B>);(<U0242>,<U0241>);/
(<U0247>,<U0246>);(<U0249>,<U0248>);(<U024B>,<U024A>);(<U024D>,<U024C>);/
(<U024F>,<U024E>);(<U0253>,<U0181>);(<U0254>,<U0186>);(<U0256>,<U0189>);/
(<U0257>,<U018A>);(<U0259>,<U018F>);(<U025B>,<U0190>);(<U0260>,<U0193>);/
(<U0263>,<U0194>);(<U0268>,<U0197>);(<U0269>,<U0196>);(<U026B>,<U2C62>);/
(<U026F>,<U019C>);(<U0272>,<U019D>);(<U0275>,<U019F>);(<U027D>,<U2C64>);/
(<U0280>,<U01A6>);(<U0283>,<U01A9>);(<U0288>,<U01AE>);(<U0289>,<U0244>);/
(<U028A>,<U01B1>);(<U028B>,<U01B2>);(<U028C>,<U0245>);(<U0292>,<U01B7>);/
(<U0345>,<U0399>);(<U037B>,<U03FD>);(<U037C>,<U03FE>);(<U037D>,<U03FF>);/
(<U03AC>,<U0386>);(<U03AD>,<U0388>);(<U03AE>,<U0389>);(<U03AF>,<U038A>);/
(<U03B1>,<U0391>);(<U03B2>,<U0392>);(<U03B3>,<U0393>);(<U03B4>,<U0394>);/
(<U03B5>,<U0395>);(<U03B6>,<U0396>);(<U03B7>,<U0397>);(<U03B8>,<U0398>);/
(<U03B9>,<U0399>);(<U03BA>,<U039A>);(<U03BB>,<U039B>);(<U03BC>,<U039C>);/
(<U03BD>,<U039D>);(<U03BE>,<U039E>);(<U03BF>,<U039F>);(<U03C0>,<U03A0>);/
(<U03C1>,<U03A1>);(<U03C2>,<U03A3>);(<U03C3>,<U03A3>);(<U03C4>,<U03A4>);/
(<U03C5>,<U03A5>);(<U03C6>,<U03A6>);(<U03C7>,<U03A7>);(<U03C8>,<U03A8>);/
(<U03C9>,<U03A9>);(<U03CA>,<U03AA>);(<U03CB>,<U03AB>);(<U03CC>,<U038C>);/
(<U03CD>,<U038E>);(<U03CE>,<U038F>);(<U03D0>,<U0392>);(<U03D1>,<U0398>);/
(<U03D5>,<U03A6>);(<U03D6>,<U03A0>);(<U03D9>,<U03D8>);(<U03DB>,<U03DA>);/
(<U03DD>,<U03DC>);(<U03DF>,<U03DE>);(<U03E1>,<U03E0>);(<U03E3>,<U03E2>);/
(<U03E5>,<U03E4>);(<U03E7>,<U03E6>);(<U03E9>,<U03E8>);(<U03EB>,<U03EA>);/
(<U03ED>,<U03EC>);(<U03EF>,<U03EE>);(<U03F0>,<U039A>);(<U03F1>,<U03A1>);/
(<U03F2>,<U03F9>);(<U03F5>,<U0395>);(<U03F8>,<U03F7>);(<U03FB>,<U03FA>);/
(<U0430>,<U0410>);(<U0431>,<U0411>);(<U0432>,<U0412>);(<U0433>,<U0413>);/
(<U0434>,<U0414>);(<U0435>,<U0415>);(<U0436>,<U0416>);(<U0437>,<U0417>);/
(<U0438>,<U0418>);(<U0439>,<U0419>);(<U043A>,<U041A>);(<U043B>,<U041B>);/
(<U043C>,<U041C>);(<U043D>,<U041D>);(<U043E>,<U041E>);(<U043F>,<U041F>);/
(<U0440>,<U0420>);(<U0441>,<U0421>);(<U0442>,<U0422>);(<U0443>,<U0423>);/
(<U0444>,<U0424>);(<U0445>,<U0425>);(<U0446>,<U0426>);(<U0447>,<U0427>);/
(<U0448>,<U0428>);(<U0449>,<U0429>);(<U044A>,<U042A>);(<U044B>,<U042B>);/
(<U044C>,<U042C>);(<U044D>,<U042D>);(<U044E>,<U042E>);(<U044F>,<U042F>);/
(<U0450>,<U0400>);(<U0451>,<U0401>);(<U0452>,<U0402>);(<U0453>,<U0403>);/
(<U0454>,<U0404>);(<U0455>,<U0405>);(<U0456>,<U0406>);(<U0457>,<U0407>);/
(<U0458>,<U0408>);(<U0459>,<U0409>);(<U045A>,<U040A>);(<U045B>,<U040B>);/
(<U045C>,<U040C>);(<U045D>,<U040D>);(<U045E>,<U040E>);(<U045F>,<U040F>);/
(<U0461>,<U0460>);(<U0463>,<U0462>);(<U0465>,<U0464>);(<U0467>,<U0466>);/
(<U0469>,<U0468>);(<U046B>,<U046A>);(<U046D>,<U046C>);(<U046F>,<U046E>);/
(<U0471>,<U0470>);(<U0473>,<U0472>);(<U0475>,<U0474>);(<U0477>,<U0476>);/
(<U0479>,<U0478>);(<U047B>,<U047A>);(<U047D>,<U047C>);(<U047F>,<U047E>);/
(<U0481>,<U0480>);(<U048B>,<U048A>);(<U048D>,<U048C>);(<U048F>,<U048E>);/
(<U0491>,<U0490>);(<U0493>,<U0492>);(<U0495>,<U0494>);(<U0497>,<U0496>);/
(<U0499>,<U0498>);(<U049B>,<U049A>);(<U049D>,<U049C>);(<U049F>,<U049E>);/
(<U04A1>,<U04A0>);(<U04A3>,<U04A2>);(<U04A5>,<U04A4>);(<U04A7>,<U04A6>);/
(<U04A9>,<U04A8>);(<U04AB>,<U04AA>);(<U04AD>,<U04AC>);(<U04AF>,<U04AE>);/
(<U04B1>,<U04B0>);(<U04B3>,<U04B2>);(<U04B5>,<U04B4>);(<U04B7>,<U04B6>);/
```



```
(<U04B9>,<U04B8>);(<U04BB>,<U04BA>);(<U04BD>,<U04BC>);(<U04BF>,<U04BE>);/
(<U04C2>,<U04C1>);(<U04C4>,<U04C3>);(<U04C6>,<U04C5>);(<U04C8>,<U04C7>);/
(<U04CA>,<U04C9>);(<U04CC>,<U04CB>);(<U04CE>,<U04CD>);(<U04CF>,<U04C0>);/
(<U04D1>,<U04D0>);(<U04D3>,<U04D2>);(<U04D5>,<U04D4>);(<U04D7>,<U04D6>);/
(<U04D9>,<U04D8>);(<U04DB>,<U04DA>);(<U04DD>,<U04DC>);(<U04DF>,<U04DE>);/
(<U04E1>,<U04E0>);(<U04E3>,<U04E2>);(<U04E5>,<U04E4>);(<U04E7>,<U04E6>);/
(<U04E9>,<U04E8>);(<U04EB>,<U04EA>);(<U04ED>,<U04EC>);(<U04EF>,<U04EE>);/
(<U04F1>,<U04F0>);(<U04F3>,<U04F2>);(<U04F5>,<U04F4>);(<U04F7>,<U04F6>);/
(<U04F9>,<U04F8>);(<U04FB>,<U04FA>);(<U04FD>,<U04FC>);(<U04FF>,<U04FE>);/
(<U0501>,<U0500>);(<U0503>,<U0502>);(<U0505>,<U0504>);(<U0507>,<U0506>);/
(<U0509>,<U0508>);(<U050B>,<U050A>);(<U050D>,<U050C>);(<U050F>,<U050E>);/
(<U0511>,<U0510>);(<U0513>,<U0512>);(<U0515>,<U0514>);(<U0517>,<U0516>);/
(<U0519>,<U0518>);(<U051B>,<U051A>);(<U051D>,<U051C>);(<U051F>,<U051E>);/
(<U0521>,<U0520>);(<U0523>,<U0522>);(<U0561>,<U0531>);(<U0562>,<U0532>);/
(<U0563>,<U0533>);(<U0564>,<U0534>);(<U0565>,<U0535>);(<U0566>,<U0536>);/
(<U0567>,<U0537>);(<U0568>,<U0538>);(<U0569>,<U0539>);(<U056A>,<U053A>);/
(<U056B>,<U053B>);(<U056C>,<U053C>);(<U056D>,<U053D>);(<U056E>,<U053E>);/
(<U056F>,<U053F>);(<U0570>,<U0540>);(<U0571>,<U0541>);(<U0572>,<U0542>);/
(<U0573>,<U0543>);(<U0574>,<U0544>);(<U0575>,<U0545>);(<U0576>,<U0546>);/
(<U0577>,<U0547>);(<U0578>,<U0548>);(<U0579>,<U0549>);(<U057A>,<U054A>);/
(<U057B>,<U054B>);(<U057C>,<U054C>);(<U057D>,<U054D>);(<U057E>,<U054E>);/
(<U057F>,<U054F>);(<U0580>,<U0550>);(<U0581>,<U0551>);(<U0582>,<U0552>);/
(<U0583>,<U0553>);(<U0584>,<U0554>);(<U0585>,<U0555>);(<U0586>,<U0556>);/
(<U1D7D>,<U2C63>);(<U1E01>,<U1E00>);(<U1E03>,<U1E02>);(<U1E05>,<U1E04>);/
(<U1E07>,<U1E06>);(<U1E09>,<U1E08>);(<U1E0B>,<U1E0A>);(<U1E0D>,<U1E0C>);/
(<U1E0F>,<U1E0E>);(<U1E11>,<U1E10>);(<U1E13>,<U1E12>);(<U1E15>,<U1E14>);/
(<U1E17>,<U1E16>);(<U1E19>,<U1E18>);(<U1E1B>,<U1E1A>);(<U1E1D>,<U1E1C>);/
(<U1E1F>,<U1E1E>);(<U1E21>,<U1E20>);(<U1E23>,<U1E22>);(<U1E25>,<U1E24>);/
(<U1E27>,<U1E26>);(<U1E29>,<U1E28>);(<U1E2B>,<U1E2A>);(<U1E2D>,<U1E2C>);/
(<U1E2F>,<U1E2E>);(<U1E31>,<U1E30>);(<U1E33>,<U1E32>);(<U1E35>,<U1E34>);/
(<U1E37>,<U1E36>);(<U1E39>,<U1E38>);(<U1E3B>,<U1E3A>);(<U1E3D>,<U1E3C>);/
(<U1E3F>,<U1E3E>);(<U1E41>,<U1E40>);(<U1E43>,<U1E42>);(<U1E45>,<U1E44>);/
(<U1E47>,<U1E46>);(<U1E49>,<U1E48>);(<U1E4B>,<U1E4A>);(<U1E4D>,<U1E4C>);/
(<U1E4F>,<U1E4E>);(<U1E51>,<U1E50>);(<U1E53>,<U1E52>);(<U1E55>,<U1E54>);/
(<U1E57>,<U1E56>);(<U1E59>,<U1E58>);(<U1E5B>,<U1E5A>);(<U1E5D>,<U1E5C>);/
(<U1E5F>,<U1E5E>);(<U1E61>,<U1E60>);(<U1E63>,<U1E62>);(<U1E65>,<U1E64>);/
(<U1E67>,<U1E66>);(<U1E69>,<U1E68>);(<U1E6B>,<U1E6A>);(<U1E6D>,<U1E6C>);/
(<U1E6F>,<U1E6E>);(<U1E71>,<U1E70>);(<U1E73>,<U1E72>);(<U1E75>,<U1E74>);/
(<U1E77>,<U1E76>);(<U1E79>,<U1E78>);(<U1E7B>,<U1E7A>);(<U1E7D>,<U1E7C>);/
(<U1E7F>,<U1E7E>);(<U1E81>,<U1E80>);(<U1E83>,<U1E82>);(<U1E85>,<U1E84>);/
(<U1E87>,<U1E86>);(<U1E89>,<U1E88>);(<U1E8B>,<U1E8A>);(<U1E8D>,<U1E8C>);/
(<U1E8F>,<U1E8E>);(<U1E91>,<U1E90>);(<U1E93>,<U1E92>);(<U1E95>,<U1E94>);/
(<U1E9B>,<U1E60>);(<U1EA1>,<U1EA0>);(<U1EA3>,<U1EA2>);(<U1EA5>,<U1EA4>);/
(<U1EA7>,<U1EA6>);(<U1EA9>,<U1EA8>);(<U1EAB>,<U1EAA>);(<U1EAD>,<U1EAC>);/
(<U1EAF>,<U1EAE>);(<U1EB1>,<U1EB0>);(<U1EB3>,<U1EB2>);(<U1EB5>,<U1EB4>);/
(<U1EB7>,<U1EB6>);(<U1EB9>,<U1EB8>);(<U1EBB>,<U1EBA>);(<U1EBD>,<U1EBC>);/
(<U1EBF>,<U1EBE>);(<U1EC1>,<U1EC0>);(<U1EC3>,<U1EC2>);(<U1EC5>,<U1EC4>);/
(<U1EC7>,<U1EC6>);(<U1EC9>,<U1EC8>);(<U1ECB>,<U1ECA>);(<U1ECD>,<U1ECC>);/
(<U1ECF>,<U1ECE>);(<U1ED1>,<U1ED0>);(<U1ED3>,<U1ED2>);(<U1ED5>,<U1ED4>);/
(<U1ED7>,<U1ED6>);(<U1ED9>,<U1ED8>);(<U1EDB>,<U1EDA>);(<U1EDD>,<U1EDC>);/
(<U1EDF>,<U1EDE>);(<U1EE1>,<U1EE0>);(<U1EE3>,<U1EE2>);(<U1EE5>,<U1EE4>);/
(<U1EE7>,<U1EE6>);(<U1EE9>,<U1EE8>);(<U1EEB>,<U1EEA>);(<U1EED>,<U1EEC>);/
(<U1EEF>,<U1EEE>);(<U1EF1>,<U1EF0>);(<U1EF3>,<U1EF2>);(<U1EF5>,<U1EF4>);/
(<U1EF7>,<U1EF6>);(<U1EF9>,<U1EF8>);(<U1F00>,<U1F08>);(<U1F01>,<U1F09>);/
(<U1F02>,<U1F0A>);(<U1F03>,<U1F0B>);(<U1F04>,<U1F0C>);(<U1F05>,<U1F0D>);/
(<U1F06>,<U1F0E>);(<U1F07>,<U1F0F>);(<U1F10>,<U1F18>);(<U1F11>,<U1F19>);/
(<U1F12>,<U1F1A>);(<U1F13>,<U1F1B>);(<U1F14>,<U1F1C>);(<U1F15>,<U1F1D>);/
(<U1F20>,<U1F28>);(<U1F21>,<U1F29>);(<U1F22>,<U1F2A>);(<U1F23>,<U1F2B>);/
(<U1F24>,<U1F2C>);(<U1F25>,<U1F2D>);(<U1F26>,<U1F2E>);(<U1F27>,<U1F2F>);/
(<U1F30>,<U1F38>);(<U1F31>,<U1F39>);(<U1F32>,<U1F3A>);(<U1F33>,<U1F3B>);/
(<U1F34>,<U1F3C>);(<U1F35>,<U1F3D>);(<U1F36>,<U1F3E>);(<U1F37>,<U1F3F>);/
(<U1F40>,<U1F48>);(<U1F41>,<U1F49>);(<U1F42>,<U1F4A>);(<U1F43>,<U1F4B>);/
(<U1F44>,<U1F4C>);(<U1F45>,<U1F4D>);(<U1F51>,<U1F59>);(<U1F53>,<U1F5B>);/
(<U1F55>,<U1F5D>);(<U1F57>,<U1F5F>);(<U1F60>,<U1F68>);(<U1F61>,<U1F69>);/
(<U1F62>,<U1F6A>);(<U1F63>,<U1F6B>);(<U1F64>,<U1F6C>);(<U1F65>,<U1F6D>);/
```



```
(<U1F66>,<U1F6E>);(<U1F67>,<U1F6F>);(<U1F70>,<U1FBA>);(<U1F71>,<U1FBB>);/
(<U1F72>,<U1FC8>);(<U1F73>,<U1FC9>);(<U1F74>,<U1FCA>);(<U1F75>,<U1FCB>);/
(<U1F76>,<U1FDA>);(<U1F77>,<U1FDB>);(<U1F78>,<U1FF8>);(<U1F79>,<U1FF9>);/
(<U1F7A>,<U1FEA>);(<U1F7B>,<U1FEB>);(<U1F7C>,<U1FFA>);(<U1F7D>,<U1FFB>);/
(<U1F80>,<U1F88>);(<U1F81>,<U1F89>);(<U1F82>,<U1F8A>);(<U1F83>,<U1F8B>);/
(<U1F84>,<U1F8C>);(<U1F85>,<U1F8D>);(<U1F86>,<U1F8E>);(<U1F87>,<U1F8F>);/
(<U1F90>,<U1F98>);(<U1F91>,<U1F99>);(<U1F92>,<U1F9A>);(<U1F93>,<U1F9B>);/
(<U1F94>,<U1F9C>);(<U1F95>,<U1F9D>);(<U1F96>,<U1F9E>);(<U1F97>,<U1F9F>);/
(<U1FA0>,<U1FA8>);(<U1FA1>,<U1FA9>);(<U1FA2>,<U1FAA>);(<U1FA3>,<U1FAB>);/
(<U1FA4>,<U1FAC>);(<U1FA5>,<U1FAD>);(<U1FA6>,<U1FAE>);(<U1FA7>,<U1FAF>);/
(<U1FB0>,<U1FB8>);(<U1FB1>,<U1FB9>);(<U1FB3>,<U1FBC>);(<U1FBE>,<U0399>);/
(<U1FC3>,<U1FCC>);(<U1FD0>,<U1FD8>);(<U1FD1>,<U1FD9>);(<U1FE0>,<U1FE8>);/
(<U1FE1>,<U1FE9>);(<U1FE5>,<U1FEC>);(<U1FF3>,<U1FFC>);(<U214E>,<U2132>);/
(<U2170>,<U2160>);(<U2171>,<U2161>);(<U2172>,<U2162>);(<U2173>,<U2163>);/
(<U2174>,<U2164>);(<U2175>,<U2165>);(<U2176>,<U2166>);(<U2177>,<U2167>);/
(<U2178>,<U2168>);(<U2179>,<U2169>);(<U217A>,<U216A>);(<U217B>,<U216B>);/
(<U217C>,<U216C>);(<U217D>,<U216D>);(<U217E>,<U216E>);(<U217F>,<U216F>);/
(<U2184>,<U2183>);(<U24D0>,<U24B6>);(<U24D1>,<U24B7>);(<U24D2>,<U24B8>);/
(<U24D3>,<U24B9>);(<U24D4>,<U24BA>);(<U24D5>,<U24BB>);(<U24D6>,<U24BC>);/
(<U24D7>,<U24BD>);(<U24D8>,<U24BE>);(<U24D9>,<U24BF>);(<U24DA>,<U24C0>);/
(<U24DB>,<U24C1>);(<U24DC>,<U24C2>);(<U24DD>,<U24C3>);(<U24DE>,<U24C4>);/
(<U24DF>,<U24C5>);(<U24E0>,<U24C6>);(<U24E1>,<U24C7>);(<U24E2>,<U24C8>);/
(<U24E3>,<U24C9>);(<U24E4>,<U24CA>);(<U24E5>,<U24CB>);(<U24E6>,<U24CC>);/
(<U24E7>,<U24CD>);(<U24E8>,<U24CE>);(<U24E9>,<U24CF>);(<U2C30>,<U2C00>);/
(<U2C31>,<U2C01>);(<U2C32>,<U2C02>);(<U2C33>,<U2C03>);(<U2C34>,<U2C04>);/
(<U2C35>,<U2C05>);(<U2C36>,<U2C06>);(<U2C37>,<U2C07>);(<U2C38>,<U2C08>);/
(<U2C39>,<U2C09>);(<U2C3A>,<U2C0A>);(<U2C3B>,<U2C0B>);(<U2C3C>,<U2C0C>);/
(<U2C3D>,<U2C0D>);(<U2C3E>,<U2C0E>);(<U2C3F>,<U2C0F>);(<U2C40>,<U2C10>);/
(<U2C41>,<U2C11>);(<U2C42>,<U2C12>);(<U2C43>,<U2C13>);(<U2C44>,<U2C14>);/
(<U2C45>,<U2C15>);(<U2C46>,<U2C16>);(<U2C47>,<U2C17>);(<U2C48>,<U2C18>);/
(<U2C49>,<U2C19>);(<U2C4A>,<U2C1A>);(<U2C4B>,<U2C1B>);(<U2C4C>,<U2C1C>);/
(<U2C4D>,<U2C1D>);(<U2C4E>,<U2C1E>);(<U2C4F>,<U2C1F>);(<U2C50>,<U2C20>);/
(<U2C51>,<U2C21>);(<U2C52>,<U2C22>);(<U2C53>,<U2C23>);(<U2C54>,<U2C24>);/
(<U2C55>,<U2C25>);(<U2C56>,<U2C26>);(<U2C57>,<U2C27>);(<U2C58>,<U2C28>);/
(<U2C59>,<U2C29>);(<U2C5A>,<U2C2A>);(<U2C5B>,<U2C2B>);(<U2C5C>,<U2C2C>);/
(<U2C5D>,<U2C2D>);(<U2C5E>,<U2C2E>);(<U2C61>,<U2C60>);(<U2C65>,<U023A>);/
(<U2C66>,<U023E>);(<U2C68>,<U2C67>);(<U2C6A>,<U2C69>);(<U2C6C>,<U2C6B>);/
(<U2C76>,<U2C75>);(<U2C81>,<U2C80>);(<U2C83>,<U2C82>);(<U2C85>,<U2C84>);/
(<U2C87>,<U2C86>);(<U2C89>,<U2C88>);(<U2C8B>,<U2C8A>);(<U2C8D>,<U2C8C>);/
(<U2C8F>,<U2C8E>);(<U2C91>,<U2C90>);(<U2C93>,<U2C92>);(<U2C95>,<U2C94>);/
(<U2C97>,<U2C96>);(<U2C99>,<U2C98>);(<U2C9B>,<U2C9A>);(<U2C9D>,<U2C9C>);/
(<U2C9F>,<U2C9E>);(<U2CA1>,<U2CA0>);(<U2CA3>,<U2CA2>);(<U2CA5>,<U2CA4>);/
(<U2CA7>,<U2CA6>);(<U2CA9>,<U2CA8>);(<U2CAB>,<U2CAA>);(<U2CAD>,<U2CAC>);/
(<U2CAF>,<U2CAE>);(<U2CB1>,<U2CB0>);(<U2CB3>,<U2CB2>);(<U2CB5>,<U2CB4>);/
(<U2CB7>,<U2CB6>);(<U2CB9>,<U2CB8>);(<U2CBB>,<U2CBA>);(<U2CBD>,<U2CBC>);/
(<U2CBF>,<U2CBE>);(<U2CC1>,<U2CC0>);(<U2CC3>,<U2CC2>);(<U2CC5>,<U2CC4>);/
(<U2CC7>,<U2CC6>);(<U2CC9>,<U2CC8>);(<U2CCB>,<U2CCA>);(<U2CCD>,<U2CCC>);/
(<U2CCF>,<U2CCE>);(<U2CD1>,<U2CD0>);(<U2CD3>,<U2CD2>);(<U2CD5>,<U2CD4>);/
(<U2CD7>,<U2CD6>);(<U2CD9>,<U2CD8>);(<U2CDB>,<U2CDA>);(<U2CDD>,<U2CDC>);/
(<U2CDF>,<U2CDE>);(<U2CE1>,<U2CE0>);(<U2CE3>,<U2CE2>);(<U2D00>,<U10A0>);/
(<U2D01>,<U10A1>);(<U2D02>,<U10A2>);(<U2D03>,<U10A3>);(<U2D04>,<U10A4>);/
(<U2D05>,<U10A5>);(<U2D06>,<U10A6>);(<U2D07>,<U10A7>);(<U2D08>,<U10A8>);/
(<U2D09>,<U10A9>);(<U2D0A>,<U10AA>);(<U2D0B>,<U10AB>);(<U2D0C>,<U10AC>);/
(<U2D0D>,<U10AD>);(<U2D0E>,<U10AE>);(<U2D0F>,<U10AF>);(<U2D10>,<U10B0>);/
(<U2D11>,<U10B1>);(<U2D12>,<U10B2>);(<U2D13>,<U10B3>);(<U2D14>,<U10B4>);/
(<U2D15>,<U10B5>);(<U2D16>,<U10B6>);(<U2D17>,<U10B7>);(<U2D18>,<U10B8>);/
(<U2D19>,<U10B9>);(<U2D1A>,<U10BA>);(<U2D1B>,<U10BB>);(<U2D1C>,<U10BC>);/
(<U2D1D>,<U10BD>);(<U2D1E>,<U10BE>);(<U2D1F>,<U10BF>);(<U2D20>,<U10C0>);/
(<U2D21>,<U10C1>);(<U2D22>,<U10C2>);(<U2D23>,<U10C3>);(<U2D24>,<U10C4>);/
(<U2D25>,<U10C5>);(<UFF41>,<UFF21>);(<UFF42>,<UFF22>);(<UFF43>,<UFF23>);/
(<UFF44>,<UFF24>);(<UFF45>,<UFF25>);(<UFF46>,<UFF26>);(<UFF47>,<UFF27>);/
(<UFF48>,<UFF28>);(<UFF49>,<UFF29>);(<UFF4A>,<UFF2A>);(<UFF4B>,<UFF2B>);/
(<UFF4C>,<UFF2C>);(<UFF4D>,<UFF2D>);(<UFF4E>,<UFF2E>);(<UFF4F>,<UFF2F>);/
(<UFF50>,<UFF30>);(<UFF51>,<UFF31>);(<UFF52>,<UFF32>);(<UFF53>,<UFF33>);/
(<UFF54>,<UFF34>);(<UFF55>,<UFF35>);(<UFF56>,<UFF36>);(<UFF57>,<UFF37>);/
```



```
   (<UFF58>,<UFF38>);(<UFF59>,<UFF39>);(<UFF5A>,<UFF3A>);/
   (<U00010428>,<U00010400>);(<U00010429>,<U00010401>);/
   (<U0001042A>,<U00010402>);(<U0001042B>,<U00010403>);/
   (<U0001042C>,<U00010404>);(<U0001042D>,<U00010405>);/
   (<U0001042E>,<U00010406>);(<U0001042F>,<U00010407>);/
   (<U00010430>,<U00010408>);(<U00010431>,<U00010409>);/
   (<U00010432>,<U0001040A>);(<U00010433>,<U0001040B>);/
   (<U00010434>,<U0001040C>);(<U00010435>,<U0001040D>);/
   (<U00010436>,<U0001040E>);(<U00010437>,<U0001040F>);/
   (<U00010438>,<U00010410>);(<U00010439>,<U00010411>);/
   (<U0001043A>,<U00010412>);(<U0001043B>,<U00010413>);/
   (<U0001043C>,<U00010414>);(<U0001043D>,<U00010415>);/
   (<U0001043E>,<U00010416>);(<U0001043F>,<U00010417>);/
   (<U00010440>,<U00010418>);(<U00010441>,<U00010419>);/
   (<U00010442>,<U0001041A>);(<U00010443>,<U0001041B>);/
   (<U00010444>,<U0001041C>);(<U00010445>,<U0001041D>);/
   (<U00010446>,<U0001041E>);(<U00010447>,<U0001041F>);/
   (<U00010448>,<U00010420>);(<U00010449>,<U00010421>);/
   (<U0001044A>,<U00010422>);(<U0001044B>,<U00010423>);/
   (<U0001044C>,<U00010424>);(<U0001044D>,<U00010425>);/
   (<U0001044E>,<U00010426>);(<U0001044F>,<U00010427>)

% The "combining" class reflects ISO/IEC 10646-1 annex B.1
% That is, all combining characters (level 2+3).
class "combining"; /
   <U0300>..<U036F>;<U0483>..<U0486>;<U0488>..<U0489>;<U0591>..<U05BD>;/
   <U05BF>;<U05C1>..<U05C2>;<U05C4>..<U05C5>;<U05C7>;<U0610>..<U061A>;/
   <U064B>..<U065E>;<U0670>;<U06D6>..<U06DC>;<U06DE>..<U06E4>;/
   <U06E7>..<U06E8>;<U06EA>..<U06ED>;<U0711>;<U0730>..<U074A>;/
   <U07A6>..<U07B0>;<U07EB>..<U07F3>;<U0901>..<U0903>;<U093C>;/
   <U093E>..<U094D>;<U0951>..<U0954>;<U0962>..<U0963>;<U0981>..<U0983>;/
   <U09BC>;<U09BE>..<U09C4>;<U09C7>..<U09C8>;<U09CB>..<U09CD>;<U09D7>;/
   <U09E2>..<U09E3>;<U0A01>..<U0A03>;<U0A3C>;<U0A3E>..<U0A42>;/
   <U0A47>..<U0A48>;<U0A4B>..<U0A4D>;<U0A51>;<U0A70>..<U0A71>;/
   <U0A75>;<U0A81>..<U0A83>;/
   <U0ABC>;<U0ABE>..<U0AC5>;<U0AC7>..<U0AC9>;<U0ACB>..<U0ACD>;/
   <U0AE2>..<U0AE3>;<U0B01>..<U0B03>;<U0B3C>;<U0B3E>..<U0B44>;/
   <U0B47>..<U0B48>;<U0B4B>..<U0B4D>;<U0B56>..<U0B57>;<U0B82>;/
   <U0BBE>..<U0BC2>;<U0BC6>..<U0BC8>;<U0BCA>..<U0BCD>;<U0BD0>;<U0BD7>;/
   <U0C01>..<U0C03>;<U0C3E>..<U0C44>;<U0C46>..<U0C48>;<U0C4A>..<U0C4D>;/
   <U0C55>..<U0C56>;<U0C62>..<U0C63>;<U0C82>..<U0C83>;<U0CBC>;/
   <U0CBE>..<U0CC4>;/
   <U0CC6>..<U0CC8>;<U0CCA>..<U0CCD>;<U0CD5>..<U0CD6>;<U0CE2>..<U0CE3>;/
   <U0D02>..<U0D03>;<U0D3E>..<U0D44>;<U0D46>..<U0D48>;<U0D4A>..<U0D4D>;/
   <U0D57>;<U0D62>..<U0D63>;<U0D82>..<U0D83>;<U0DCA>;<U0DCF>..<U0DD4>;/
   <U0DD6>;<U0DD8>..<U0DDF>;<U0DF2>..<U0DF3>;<U0E31>;<U0E34>..<U0E3A>;/
   <U0E47>..<U0E4E>;<U0EB1>;<U0EB4>..<U0EB9>;<U0EBB>..<U0EBC>;/
   <U0EC8>..<U0ECD>;<U0F18>..<U0F19>;<U0F35>;<U0F37>;<U0F39>;/
   <U0F3E>..<U0F3F>;<U0F71>..<U0F84>;<U0F86>..<U0F87>;<U0F90>..<U0F97>;/
   <U0F99>..<U0FBC>;<U0FC6>;<U102B>..<U103F>;/
   <U1056>..<U1059>;<U105E>..<U1060>;<U1062>..<U1064>;<U1067>..<U106D>;/
   <U1071>..<U1074>;<U1082>..<U108D>;<U108F>;/
   <U135F>;<U1712>..<U1714>;<U1732>..<U1734>;/
   <U1752>..<U1753>;<U1772>..<U1773>;<U17B6>..<U17D3>;<U17DD>;/
   <U180B>..<U180D>;<U18A9>;<U1920>..<U192B>;<U1930>..<U193B>;/
   <U19B0>..<U19C0>;<U19C8>..<U19C9>;<U1A17>..<U1A1B>;<U1B00>..<U1B04>;/
   <U1B34>..<U1B44>;<U1B6B>..<U1B73>;<U1DC0>..<U1DE6>;<U1DFE>..<U1DFF>;/
   <U20D0>..<U20F0>;<U2DE0>..<U2DFF>;<U302A>..<U302F>;<U3099>..<U309A>;/
   <UA66F>..<UA672>;<UA67C>;<UA67D>;<UA802>;<UA806>;/
   <UA80B>;<UA823>..<UA827>;<UFB1E>;<UFE00>..<UFE0F>;<UFE20>..<UFE26>;/
   <U00010A01>..<U00010A03>;<U00010A05>..<U00010A06>;/
   <U00010A0C>..<U00010A0F>;<U00010A38>..<U00010A3A>;<U00010A3F>;/
   <U0001D165>..<U0001D169>;<U0001D16D>..<U0001D172>;/
   <U0001D17B>..<U0001D182>;<U0001D185>..<U0001D18B>;/
   <U0001D1AA>..<U0001D1AD>;<U0001D242>..<U0001D244>;/
```



```
   <U000E0100>..<U000E01EF>

% The "combining_level3" class reflects ISO/IEC 10646-1 annex B.2
% That is, combining characters of level 3.
class "combining_level3"; /
   <U0334>..<U0338>;<U034F>;<U0488>..<U0489>;<U05B0>..<U05BD>;<U05BF>;/
   <U05C1>..<U05C2>;<U05C7>;<U064B>..<U0652>;<U0670>;<U06DE>;<U0711>;/
   <U07A6>..<U07B0>;<U0901>..<U0903>;<U093C>;<U093E>..<U094D>;/
   <U0962>..<U0963>;<U0981>..<U0983>;<U09BC>;<U09BE>..<U09C4>;/
   <U09C7>..<U09C8>;<U09CB>..<U09CD>;<U09D7>;<U09E2>..<U09E3>;/
   <U0A01>..<U0A03>;<U0A3C>;<U0A3E>..<U0A42>;<U0A47>..<U0A48>;/
   <U0A4B>..<U0A4D>;<U0A51>;<U0A70>..<U0A71>;<U0375>;<U0A81>..<U0A83>;/
   <U0ABC>;/
   <U0ABE>..<U0AC5>;<U0AC7>..<U0AC9>;<U0ACB>..<U0ACD>;<U0AE2>..<U0AE3>;/
   <U0B01>..<U0B03>;<U0B3C>;<U0B3E>..<U0B44>;<U0B47>..<U0B48>;/
   <U0B4B>..<U0B4D>;<U0B56>..<U0B57>;<U0B82>;<U0BBE>..<U0BC2>;/
   <U0BC6>..<U0BC8>;<U0BCA>..<U0BCD>;<U0BD0>;<U0BD7>;<U0C01>..<U0C03>;/
   <U0C3E>..<U0C44>;<U0C46>..<U0C48>;<U0C4A>..<U0C4D>;<U0C55>..<U0C56>;/
   <U0C62>..<U0C63>;/
   <U0C82>..<U0C83>;<U0CBC>;<U0CBE>..<U0CC4>;<U0CC6>..<U0CC8>;/
   <U0CCA>..<U0CCD>;<U0CD5>..<U0CD6>;<U0CE2>..<U0CE3>;<U0D02>..<U0D03>;/
   <U0D3E>..<U0D44>;<U0D46>..<U0D48>;<U0D4A>..<U0D4D>;<U0D57>;/
   <U0D62>..<U0D63>;/
   <U0D82>..<U0D83>;<U0DCA>;<U0DCF>..<U0DD4>;<U0DD6>;<U0DD8>..<U0DDF>;/
   <U0DF2>..<U0DF3>;<U0E31>;<U0E34>..<U0E3A>;<U0E47>..<U0E4E>;<U0EB1>;/
   <U0EB4>..<U0EB9>;<U0EBB>..<U0EBC>;<U0EC8>..<U0ECD>;<U0F3E>..<U0F3F>;/
   <U0F71>..<U0F81>;<U0F84>;<U0F90>..<U0F97>;<U0F99>..<U0FBC>;/
   <U102B>..<U103F>;<U1056>..<U1059>;<U105E>..<U1060>;<U1062>..<U1064>;/
   <U1067>..<U106D>;/
   <U1071>..<U1074>;<U1082>..<U108D>;<U108F>;<U1712>..<U1714>;/
   <U1732>..<U1734>;<U1752>..<U1753>;<U1772>..<U1773>;<U17B6>..<U17D3>;/
   <U180B>..<U180D>;<U1920>..<U192B>;<U1930>..<U1938>;<U19B0>..<U19C0>;/
   <U19C8>..<U19C9>;<U1A19>..<U1A1B>;<U1B00>..<U1B04>;<U1B34>..<U1B44>;/
   <U20D2>..<U20D3>;<U20D8>..<U20DA>;<U20DD>..<U20E0>;<U20E2>..<U20E6>;/
   <U20EA>..<U20EB>;<U3099>..<U309A>;<UA802>;<UA806>;<UA80B>;/
   <UA823>..<UA827>;<UFB1E>;<UFE00>..<UFE0F>;<U00010A01>..<U00010A03>;/
   <U00010A05>..<U00010A06>;<U00010A0C>;<U00010A0E>;<U00010A39>;<U00010A3F>;/
   <U0001D167>..<U0001D169>;<U000E0100>..<U000E01EF>

END LC_CTYPE
```


4.4 LC_COLLATE

A collation sequence definition defines the relative order between collating elements (characters and multicharacter collating elements) in the FDCC-set. This order is expressed in terms of collation values; i.e., by assigning each element one or more collation values (also known as collation weights). This does not imply that applications assign such values, but that ordering of strings using the resultant collation definition in the FDCC-set behaves as if such assignment is done and used in the collation process. The collation sequence definition is used by regular expressions, pattern matching. When no weights are specified the collation sequence definition also is used for sorting, else the weighting defines the sorting. The following capabilities are provided:

(1) Multicharacter collating elements. Specification of multicharacter collating elements (i.e., sequences of two or more characters to be collated as an entity).

(2) User-defined ordering of collating elements. Each collating element is assigned a collation value defining its order in the character (or basic) collation sequence. This ordering is used by regular expressions and pattern matching and, unless collation weights are explicitly specified, also as the collation weight to be used in sorting.

(3) Multiple weights and equivalence classes. Collating elements can be assigned one or more (up to the limit (COLL_WEIGHTS_MAX)) collating weights for use in sorting. The first weight is hereafter referred to as the primary weight.

(4) One-to Many mapping. A single character is mapped into a string of collating elements.

(5) Many-to-Many substitution. A string of one or more characters is substituted by another string (or an empty string, i.e., the character or characters are ignored for collation purposes).

(6) Equivalence class definition. Two or more collating elements have the same collation value (primary weight).

(7) Ordering by weights. When two strings are compared to determine their relative order, the two strings are first broken up into a series of collating elements, and each successive pair of elements are compared according to the relative primary weights for the elements. If equal, and more than one weight has been assigned, then the pairs of collating elements are recompared according to the relative subsequent weights, until either a pair of collating elements compare unequal or the weights are exhausted.

(8) Easy reordering of characters. ISO/IEC 14651 has a template for collation specification that with just a few modifications can be culturally correct for a specific culture. Here the "reorder-after" keyword gives a convenient way to modify a FDCC-set template.

(9) Easy reordering of sections. The template in ISO/IEC 14651 gives an ordering of the sections that may not be culturally acceptable in certain cultures. The keyword "reorder-section-after" gives a convenient way to modify the order of sections in a FDCC-set

template.

The following keywords are recognized in a collation sequence definition. Some of them are described in detail in the following subclauses. The keywords are mandatory unless otherwise noted.

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. The FDCC-set is copied in source form.
coll_weight_max	Define as a decimal number the number of collation levels that an interpreting system needs to support for this FDCC-set, this value is elsewhere referred to as the COLL_WEIGHT_MAX limit (e.g. in the "order_start" statement). An interpreting system caters for up to 7 collating levels.
section-symbol	Define a section symbol representing a set of collation order statements. The section is defined with the "order_start" keyword until the next "order_start" or "order_end" keyword. This keyword is optional.
collating-element	Define a collating-element symbol representing a multicharacter collating element. This keyword is optional.
collating-symbol	Define one or more collating symbols for use in collation order statements. This keyword is optional.
symbol-equivalence	Define a collating-symbol to be equivalent to another defined collating-symbol.
order_start	Define collation rules. This statement is followed by one or more collation order statements, assigning character collation values and collation weights to collating elements.
order_end	Specify the end of the collation-order statements.
section	Specify a section of collation order statements, and optionally a subrepertoire thereof.
reorder-after	Redefine collating rules. Specify after which collating element the redefinition of collation order takes order. This statement is followed by one or more collation order statements, reassigning character collation values and collation weights to collating elements.
reorder-end	Specify the end of the "reorder-after" collating order statements.
reorder-section-after	Redefine the order of sections. This statement is followed by one or more section symbols, reassigning character collation values and collation weights to collating elements.
reorder-section-end	Specify the end of the "reorder-section" section order statements.

4.4.1 Collation statements

The "order_start", "reorder-after" and "section" keywords are followed by collating statements. The syntax for the collating statements is

"%s %s;%s;...;%s\n",<collating-identifier>,<weight>,<weight>,...

Each <collating-identifier> consists of either a character (in any of the forms defined in 4.1.1), a <collating-element>, a <collating-symbol>, an ellipsis, or the special symbol "UNDEFINED". The weights for each of the collation elements determines the character collation sequence - such that each collation statement does not need to be in collation order, and weights could be rearranged via for example the "reorder-after" keyword. No character has any specific predetermined placement in the collation sequence. The order in which collating elements are specified determines the character collation sequence, such that each collating element compares less than the elements following it.

A <collating-element> is used to specify multicharacter collating elements, and indicates that the character sequence specified via the <collating-element> is to be collated as a unit and in the relative order specified by its place in the list of collating statements.

A <collating-symbol> is used to define a position in the relative order for use in weights.

The absolute ellipsis symbol ("...") specifies that a sequence of characters collate according to their encoded character values. It is interpreted as indicating that all characters with a coded character set value higher than the value of the character in the preceding line, and lower than the coded character set value for the character in the following line, in the current coded character set, are placed in the character collation order between the previous and the following character in ascending order according to their coded character set values. An initial ellipsis is interpreted as if the preceding line specified the <NUL> character, and a trailing ellipsis as if the following line specified the highest coded character set value in the current coded character set. An ellipsis is treated as invalid if the preceding or following lines do not specify characters in the current coded character set. The use of the ellipsis symbol ties the definition to a specific coded character set and may preclude the definition from being portable between applications, and is depreciated. Symbolic ellipses may be used as the ellipses symbol, but generating symbolic character names, and thus have a better chance of portability between applications.

The symbolic ellipses (".." or "....") specifies a sequence of collating statements. It is interpreted as indicating that all characters with symbolic names higher than the symbolic name of the character in the preceding line, and lower in the sequence of symbolic names for the character in the following line, is placed in the character collation order between the previous and the following character

in ascending order.

The symbol "UNDEFINED" is interpreted as including all coded character set values not specified explicitly or via the ellipsis or one of the symbolic ellipses symbols. Such characters are inserted in the character collation order at the point indicated by the symbol, and in ascending order according to their coded character set values. If no "UNDEFINED" symbol is specified, and the current coded character set contains characters not specified in this clause, the utility issues a warning message and place such characters at the end of the character collation order.

The optional operands for each collation-element are used to define the primary, secondary, or subsequent weights for the collating element. The first operand specifies the relative primary weight, the second the relative secondary weight, and so on. Two or more collation-elements can be assigned the same weight; they belong to the same equivalence class if they have the same primary weight. Collation behaves as if, for each weight level, "IGNORE"d elements are removed. Then each successive pair of elements is compared according to the relative weights for the elements. If the two strings compare equal, the process is repeated for the next weight level, up to the limit "COLL_WEIGHTS_MAX" of the associated FDCC-set.

Weights are expressed as characters (in any of the forms specified here), <collating-symbol>s, <collating-element>s, an ellipsis, or the special symbol "IGNORE". A single character, a <collating-symbol>, or a <collating-element> represent the relative order in the character collating sequence of the character or symbol, rather than the character or characters themselves.

One-to-many mapping is indicated by specifying two or more concatenated characters or symbolic names. Thus, if the character <ss> is given the string <s><s> as a weight, comparisons are performed as if all occurrences of the character <ss> are replaced by <s><s>. If it is desirable to define <ss> and <s><s> as an equivalence class, then a collating-element must be defined for the string "ss", as in the example below.

All characters specified via an ellipsis are by default assigned unique weights, equal to the relative order of characters. Characters specified via an explicit or implicit "UNDEFINED" special symbol are by default assigned the same primary weight (i.e., belong to the same equivalence class). An ellipsis symbol as a weight is interpreted to mean that each character in the sequence has unique weights, equal to the relative order of their character in the character collation sequence. Secondary and subsequent weights have unique values. The use of the ellipsis as a weight is treated as an error if the collating element is neither an ellipsis nor the special symbol "UNDEFINED".

The special keyword "IGNORE" as a weight indicates that when strings are compared using the weights at the level where "IGNORE" is specified, the

collating element is ignored; i.e., as if the string did not contain the collating element. In regular expressions and pattern matching, all characters that are "IGNORE"d in their primary weight form an equivalence class.

A <comment_character> occurring where the delimiter ";" may occur, terminates the collating statement.

An empty operand is interpreted as the collating-element itself.

For example, the collation statement

```
<a>  <a>;<a>
```

is equal to

```
<a>
```

An ellipsis (absolute or symbolic) can be used as an operand if the collating-element was an ellipsis, and is interpreted as the value of each character defined by the ellipsis.

```
Example:

collating-element <ch> from "<c><h>"
collating-element <Ch> from "<C><h>"
order_start    forward;backward
UNDEFINED      IGNORE;IGNORE
<LOW>
<space>        <LOW>;<space>
...            <LOW>;
<a>            <a>;<a>
<a'>           <a>;<a'>
<A>            <a>;<A>
<A'>           <a>;<A'>
<ch>           <ch>;<ch>
<Ch>           <ch>;<Ch>
<s>            <s>;<s>
<ss>           "<s><s>";"<ss><ss>"
order_end
```

This example is interpreted as follows:

(1) The UNDEFINED means that all characters not specified in this definition (explicitly or via the ellipsis) is ignored.
(2) <LOW> defines the first collating weight, and thus the lowest weight in this example.
(3) All characters between <space> and <a> have the same primary equivalence class <LOW> and individual secondary weights based on their ordinal encoded values. (The use of absolute ellipses is depreciated, but used here to illustrate generic use of ellipses. Symbolic ellipses should be used instead).
(4) All characters based on the upper or lowercase character "a" belong to the same primary equivalence class.
(5) The multicharacter collating element <c><h> is represented by the collating symbol <ch> and belongs to the same primary equivalence class as the multicharacter collating element <C><h>.
(6) The <ss> collating element has two weights on the primary level, and it is in the same primary equivalence class as two consecutive <s>-es; on the secondary level the collating element has two weights of the equivalence class <ss>.

4.4.2 "copy" keyword

This keyword specifies the name of an existing FDCC-set to be used as the source for the definition of this category. The syntax is

"copy %s\n", <FDCC-set-name>

The <FDCC-set-name> consists of one or more characters (in any of the forms defined in 4.1.1). The FDCC-set is copied in source form.

4.4.3 "coll_weight_max" keyword

This keyword defines as a decimal number the number of collation levels that an interpreting system needs to support. An interpreting system caters for up to 7 collating levels. The syntax is

"coll_weight_max %d\n", <value>

4.4.4 "section-symbol" keyword

This keyword is used to define symbols for use in section related statements; such as the "order_start", and "reorder-section-after" keywords and section-reordering statements. The syntax is

"section-symbol %s\n", <section-symbol>

The <section-symbol> is a symbolic name, enclosed between angle brackets (< and >), and does not duplicate any symbolic name in the current charmap (if any), or any other symbolic name defined in this collation definition. A <section-symbol> defined via this keyword is only defined within the LC_COLLATE category.

Example:
```
section-symbol <LATIN>
section-symbol <ARABIC>
```

4.4.5 "collating-element" keyword

In addition to the collating elements in the character set, the collating-element keyword is used to define multicharacter collating elements. The syntax is

"collating-element %s from %s\n",<collating-symbol>,<string>

The <collating-symbol> operand is a symbolic name, enclosed between angle brackets (< and >), and does not duplicate any symbolic name in the current charmap or repertoiremap file (if any), or any other symbolic name defined in

this collation definition. The string operand is a string of two or more characters that collates as an entity. A <collating-element> defined via this keyword is only defined within the LC_COLLATE category.

Example with ISO/IEC 10646:
collating-element <ch> from "<c><h>"
collating-element <e-acute> from "<e><combining-acute>"
collating-element <aa> from "<a><a>"

Note: The problem of comparing a fully composed character of ISO/IEC 10646 with a decomposed representation of the same text is sometimes handled by the two strings comparing equal up to level 3 (the case level) of ISO/IEC 14651, but distinguishing the two at the 4th level.

4.4.6 "collating-symbol" keyword

This keyword is used to define symbols for use in collation sequence statements; e.g., between the order_start and the order_end keywords. The syntax is

"collating-symbol %s;%s;...%s\n", <collating-symbol>, <collating-symbol> ...

The <collating-symbol> is a symbolic name, enclosed between angle brackets (< and >), and does not duplicate any symbolic name in the current charmap (if any), or any other symbolic name defined in this collation definition. A <collating-symbol> defined via this keyword is only defined within the LC_COLLATE category. More than one <collating-symbol> may be defined with one "collating-symbol" keyword, and symbolic ellipses may be used.

Example:
collating-symbol <CAPITAL>
collating-symbol <HIGH>

4.4.7 "symbol-equivalence" keyword

This keyword is used to define symbols for use in collation sequence statements; and assign the same weight as another defined symbol. The syntax is

"symbol-equivalence %s %s\n", <collating-symbol-1>, <collating-symbol-2>

The <collating-symbol-1> and <collating-symbol-2> are symbolic names, enclosed between angle brackets (< and >). <collating-symbol-1> does not duplicate any symbolic name in the current charmap (if any), or any other symbolic name defined in this collation definition. <collating-symbol-2> is defined elsewhere in the LC_COLLATE category as a collating-symbol. The use of <collating-symbol-2> is equivalent to using the <collating-symbol-1> in the LC_COLLATE category. A <collating-symbol-1> defined via this keyword is only defined within the LC_COLLATE category.

Example
collating-symbol <CAP>
symbol-equivalence <CAPITAL> <CAP>

4.4.8 "order_start" keyword

The "order_start" keyword precedes collation order entries and also defines the number of weights for this collation sequence definition, the collation section name and other collation rules.

The syntax of the "order_start" keyword has two forms:

"order_start %s;%s;...;%s\n", <sort-rule>, <sort-rule> ...
and
"order_start %s;%s;...;%s\n", <section-symbol>, <sort-rules>, <sort-rules> ...

The operands to the order_start keyword are optional. If present, the operands define rules to be applied when strings are compared. The first operand may be a <section-symbol> surrounded by "<" and ">" and the set of collating statements following the "order_start" keyword until the "order_end" keyword are identified with this <section-symbol> or another "order_start" keyword is encountered. The remaining number of operands define how many weights each element is assigned; if no operands are present, one forward operand is assumed. If present, the first operand defines rules to be applied when comparing strings using the first (primary) weight; the second when comparing strings using the second weight, and so on. Operands are separated by semicolons (;). Each operand consists of one or more collation directives, separated by commas (,). If the number or operands exceeds the (COLL_WEIGHTS_MAX) limit, a utility parsing the FDCC-set description issues a warning message. The following directives are supported:

forward	Specifies that the direction of scanning a part of a string at a given point in a string is done towards the logical end of the whole string for this weight level.
backward	Specifies that the direction of scanning a part of a string at a given point in a string is done towards the logical beginning of the whole string for this weight level.
position	Specifies that comparison operations for the weight level will consider the relative position of non-"IGNORE"d elements in the strings. The string containing a non-"IGNORE"d element after the fewest IGNOREd collating elements from the start of the compare collates first. If both strings contain a non-"IGNORE"d character in the same relative position, the collating values assigned to the elements determine the ordering. In case of equality, subsequent non-

IGNOREd characters are considered in the same manner.

The directives "forward" and "backward" are mutually exclusive at a given level. The directives "backward" and "position" are mutually exclusive at a given level.

Examples:
order_start forward;backward
order_start <CYRILLIC>;forward;forward

If no operands are specified, a single forward operand is assumed.

4.4.9 "order_end" keyword

The collating order entries are terminated with an "order_end" keyword.

4.4.10 "reorder-after" keyword

The "reorder-after" keyword is used to specify a modification to a copied collation specification of an existing FDCC-set. There can be more than one "reorder-after" statement in a collating specification. The syntax is:

"reorder-after %s\n",<collating-symbol>

The <collating-symbol> operand is a symbolic name, enclosed between angle brackets, and is present in the source FDCC-set copied via the "copy" keyword. The "reorder-after" statement is followed by one or more collation statements as described in the "Collating Order" clause (4.4.5), with the exception that the ellipsis symbol (...) is not used.

Each collation statement reassigns character collation values and collation weights to collating elements existing in the copied collation specification, by removing the collating statement from the copied specification, and inserting the collating element in the collating sequence with the new collation weights after the preceding collating element of the "reorder-after" specification, the first collating element in the collation sequence being the <collating-symbol> specified in the "reorder-after" statement.

A "reorder-after" specification is terminated by another "reorder-after" specification or the "reorder-end" statement.

4.4.10.1 Example of "reorder-after"

```
reorder-after <y8>
<U:>        <Y>;<U:>;<CAPITAL>
<u:>        <Y>;<U:>;<SMALL>
reorder-after <z8>
<AE>        <AE>;<NONE>;<CAPITAL>
<ae>        <AE>;<NONE>;<SMALL>
```



```
<A:>        <AE>;<DIAERESIS>;<CAPITAL>
<a:>        <AE>;<DIAERESIS>;<SMALL>
<O/>        <O/>;<NONE>;<CAPITAL>
<o/>        <O/>;<NONE>;<SMALL>
<AA>        <AA>;<NONE>;<CAPITAL>
<aa>        <AA>;<NONE>;<SMALL>
reorder-end
```

The example is interpreted as follows (using the "i18nrep" repertoiremap):

1. The collating element <U:> is removed from the copied collating sequence and inserted after <y8> in the collating sequence with the new weights. The collating element <u:> is removed from the copied collating sequence and inserted in the resulting collation sequence after <U:> with the new weights. <y8> is used to indicate the position of the last y letter.

2. The second "reorder-after" statement terminates the first list of reordering collation identifier entries, and initiates a second list, rearranging the order and weights for the <AE>, <ae>, <A:>, <a:>, <O/>, and <o/> collating elements after the <z8> collating symbol in the copied specification. <z8> is used to indicate the position of the last z letter.

3. The "reorder-end" statement terminates the second list of reordering entries.

4. Thus for the original sequence

   ... (U u Ü ü) V v W w X x Y y Z z

   this example reordering gives

   ... U u V v W w X x (Y y Ü ü) Z z (Æ æ Ä ä) Ø ø Å å

   where the parenthesis indicate ordering with the same weight on the first level for multiple upper/lowercase pairs.

4.4.11 "reorder-end" keyword

The "reorder-end" keyword specifies the end of a list of collating statements, initiated by the "reorder-after" keyword.

4.4.12 "section" keyword

The "section" keyword is used to define a section of the table. A section consists of a set of collation elements with their associated collation weights. A section can be moved as a whole via the "reorder-section-after" keyword.

Each "section" keyword has the syntax:

"section %s %s;...;%s\n", <section-symbol>, <collation-symbol>, ...

The <section-symbol> is a symbolic name, enclosed between angle brackets "<" and ">", and it defines the name of the section in question. It may have been defined in a "section-symbol" statement.

The <collation-symbol> is a symbolic name, enclosed between angle brackets "<" and ">", and it references a collating element previously specified, with associated weights. More than one <collating-symbol> may be referenced in one "section" statement, and symbolic ellipses may be used. The <collation-symbol>s identified via this list are removed from other parts of the collation specification. The list of <collation-symbol>s is optional.

A section consists of the collating elements identified on the "section" keyword line and with relative order and weights as specified earlier, plus the collation elements defined via the optionally following collating statements as described in 4.4.1. The section is terminated by another keyword line.

4.4.13 "reorder-section-after" keyword

The "reorder-section-after" keyword is used to specify a modification to a copied collation specification of an existing FDCC-set. The "reorder-section-after" statement is followed by one or more statements consisting of section reordering statements.

Each "reorder-section-after" keyword has either the syntax:

"reorder-section-after %s\n", <collation-symbol>
or:
"reorder-section-after %s %s", <section-symbol>, <collation-symbol>

The <collation-symbol> is a symbolic name, enclosed between angle brackets "<" and ">", and it references a collating element previously specified.

The <section-symbol> is a symbolic name, enclosed between angle brackets "<" and ">", and it refers to the name of the section in question, previously defined in a "section-symbol" or "section" keyword, and with contents allocated via a "order_start" or "section" keyword.

If there is no <section-symbol> given with the keyword, the keyword is followed by a number of section reordering statements, terminated by a "reorder-section-end" keyword.

The collating elements and associated weights of the section given with the keyword line, or the sections given on the following lines, are removed from the current sorting table, possibly reassigned sorting rules according to the section reordering statements, and inserted in the sorting table after the <collating-symbol>.

4.4.13.1 section reordering statements

The section reordering statements rearranges the set of collating entries and changes sorting rules for the set of collating entries identified by a section symbol in a preceding "order_start" statement. Each section reorder statement has the syntax:

"%s %s;...%s\n", <section-symbol>, <sort-rule>, <sort-rule> ...

The <section-symbol> identifies the set of collating entries. The <section-symbol> is defined via a "section-symbol" or the "section" keyword, and values identified by the <section-symbol> is assigned via the "order_start" or "section" keywords.

The <sort-rule>s are as described for the "order_start" keyword. Specified <sort-rule>s replace the specification of the ordering given on the first

"order_start" statement, for the section identified by the <section-symbol> . The <sort-rule>s are optional, and <sort-rule>s not to be changed from the first "order_start" specification is given by empty specifications on the "section" statement.

Note: The <sort-rule> capability is an extension over ISO/IEC 14651 functionality.

The order of the section reordering statements rearranges the assignment of collation entries for the sets of collation entries identified by the <section-symbols> to the order that the <section-symbols> occur after the "reorder-section-after" statement.

The section reordering statements are terminated by a "reorder-section-end" statement.

4.4.13.2 Example of section reordering

```
copy "i18n"
section <DEVANAGARI> <U0905>..<U0939>;<U093D>..<U0950>
reorder-section-after <DEVANAGARI> <U3361>
```

This example is interpreted as follows: The LC_COLLATE category of the "i18n" FDCC-set is copied. Then a definition of the section <DEVANAGARI> is done, and the collating elements of this section is removed from the table and inserted in the same relative order and with the same weights after the collating element <U3361>, which is the last of the digits. In this way the <DEVANAGARI> section is reordered to be sorted before all other letters.

4.4.14 "reorder-section-end" keyword

The "reorder-section-end" keyword specifies the end of a list of section symbols, initiated by the "reorder-section-after" keyword.

4.4.15 "i18n" LC_COLLATE category

The "i18n" LC_COLLATE category is defined as the following, which includes the tailorable template in ISO/IEC 14651.

```
LC_COLLATE
% This is the ISO/IEC TR 30112 i18n fdcc-set definition for
% the LC_COLLATE category.
%
% equivalences
symbol-equivalence <NONE>              <BASE>
symbol-equivalence <CAPITAL>           <CAP>
symbol-equivalence <SMALL>             <MIN>
symbol-equivalence <CAPITAL-SMALL>     <COMPATCAP>
symbol-equivalence <SMALL-CAPITAL>     <COMPAT>
symbol-equivalence <MACRON>            <MACRO>
symbol-equivalence <STROKE>            <OBLIK>
symbol-equivalence <ACUTE>             <AIGUT>
symbol-equivalence <CIRCUMFLEX>        <CIRCF>
symbol-equivalence <RING>              <CRCLE>
symbol-equivalence <DIAERESIS>         <TREMA>
symbol-equivalence <DOT>               <POINT>
```



```
symbol-equivalence <CEDILLA>           <CEDIL>
symbol-equivalence <OGONEK>            <OGONK>
symbol-equivalence <HOOK>              <CROOK>
symbol-equivalence <HORN>              <HORNU>
symbol-equivalence <DOT-BELOW>         <POINS>

% Copy the template from ISO/IEC 14651
copy "ISO14651_2006_TABLE1_en.txt"
reorder-after <SFFFF>
order_start forward;forward;forward;forward,position
reorder-end
END LC_COLLATE
```


4.5 LC_MONETARY

The LC_MONETARY category defines the rules and symbols that are used to format monetary numeric information. The operands are strings. For some keywords, the strings can contain only integers. More than one set of monetary values may be provided, and for each set a period of validity and conversion rate may be given. Keywords that are not provided, string values set to the empty string "", or integer keywords set to -1, are used to indicate that the value is unspecified, and then no default is implied. The following keywords are defined:

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.
valid_from	One or more strings separated by semicolons, representing a Gregorian date in the form "YYYYMMDD" according to ISO 8601, specifying the beginning date (inclusive from the beginning of day local time) of the validity of a currency. The position of the string in the list corresponds to the position of operands in other keywords in the LC_MONETARY category. The currencies should be ordered in terms of validity dates, and for each validity period with the currency that the amounts are stored in first. If not specified, it is taken to be an implementation-defined beginning of time. This keyword is optional.
valid_to	One or more strings, separated by semicolons, each representing a Gregorian date in the form "YYYYMMDD" according to ISO 8601, that specify the last date (inclusive to the end of day local time) of the validity of a currency. If not specified, it is taken to be an implementation-defined end of time. This keyword is optional.
conversion_rate	one or more pairs of integers separated by a <semicolon> specifying the fixed conversion rate between the current currency and the first currency that is valid, determined by a date provided by the application. If the currency is not the first valid currency for the period in question, the first integer is for multiplying the first valid currency, and the second for dividing this result to get the amount in the current currency. The currency to be the current currency is selected by the application from the date applicable; and whether domestic or international formatting is used is also determined by the application. Each pair of integers are separated by a <slash>. The default value is "1/100". This keyword is optional.

Note: The two integers are used instead of a floating point value, to be

able to cater for legal requirements on Euro conversion where a multiplication and division is prescribed, instead of just one floating point multiplication.

currency_symbol One or more strings separated by semicolons that are used as the local currency symbol.
Note: The symbols normally identify different currencies.

mon_decimal_point The operand is one or more strings separated by semicolons containing the symbol that is used as the decimal delimiter in monetary formatted quantities. In contexts where other standards limit the "mon_decimal_point" to a single byte, the result of specifying a multibyte operand is unspecified. The keyword is specified, unless the "copy" keyword is used.

mon_thousands_sep The operand is one or more strings separated by semicolons containing the symbol that is used as a separator for groups of digits to the left of the decimal delimiter in formatted monetary quantities. In contexts where other standards limit the "mon_thousands_sep" to a single byte, the result of specifying a multibyte operand is unspecified. The keyword is specified, unless the "copy" keyword is used.

mon_grouping Define the size of each group of digits in formatted monetary quantities. The operand is a sequence of integers separated by semicolons. Each integer specifies the number of digits in each group, with the initial integer defining the size of the group immediately preceding the decimal delimiter, and the following integers defining the preceding groups. If the last integer is not -1, then the size of the previous group (if any) is repeatedly used for the remainder of the digits. If the last integer is -1, then no further grouping is performed. The keyword is specified, unless the "copy" keyword is used.

positive_sign A string that is used to indicate a non-negative-valued formatted monetary quantity. The keyword is specified, unless the "copy" keyword is used.

negative_sign A string that is used to indicate a negative-valued formatted monetary quantity. The keyword is specified, unless the "copy" keyword is used.

frac_digits One or more integers separated by semicolons, representing the number of fractional digits (those to the right of the decimal delimiter) to be written in a formatted monetary quantity using "currency_symbol". The keyword is specified, unless the "copy" keyword is used.

p_cs_precedes One or more integers separated by semicolons, set to 1 if the "currency_symbol" precedes the value for a non-negative formatted monetary quantity, and set to 0 if

	the symbol succeeds the value. The keyword is specified, unless the "copy" keyword is used.
p_sep_by_space	One or more integers separated by semicolons, set to 0 if no space separates the "currency_symbol" from the value for a non-negative formatted monetary quantity, set to 1 if a space separates the symbol from the value, and set to 2 if a space separates the symbol and the sign string, if adjacent. The keyword is specified, unless the "copy" keyword is used.
n_cs_precedes	One or more integers separated by semicolons, set to 1 if the "currency_symbol" precedes the value for a negative formatted monetary quantity, and set to 0 if the symbol succeeds the value. The keyword is specified, unless the "copy" keyword is used.
n_sep_by_space	One or more integers separated by semicolons, set to 0 if no space separates the "currency_symbol" from the value for a negative formatted monetary quantity, set to 1 if a space separates the symbol from the value, and set to 2 if a space separates the symbol and the sign string, if adjacent. The keyword is specified, unless the "copy" keyword is used.
p_sign_posn	One or more integers separated by semicolons, set to a value indicating the positioning of the "positive_sign" for a non-negative formatted monetary quantity using the "currency_symbol". The following integer values are defined: 0 Parentheses enclose the quantity and the "currency_symbol". 1 The sign string precedes the quantity and the "currency_symbol". 2 The sign string succeeds the quantity and the "currency_symbol". 3 The sign string immediately precedes the "currency_symbol". 4 The sign string immediately succeeds the "currency_symbol". The keyword is specified, unless the "copy" keyword is used.
n_sign_posn	One or more integers separated by semicolons, set to a value indicating the positioning of the "negative_sign" for a negative formatted monetary quantity using the "currency_symbol". The following integer values are defined: 0 Parentheses enclose the quantity and the "currency_symbol". 1 The sign string precedes the quantity and the "currency_symbol". 2 The sign string succeeds the quantity and the "currency_symbol". 3 The sign string immediately precedes the "currency_symbol". 4 The sign string immediately succeeds the "currency_symbol". The keyword is specified, unless the "copy" keyword is used.
int_curr_symbol	One or more strings separated by semicolons that are used as the international currency symbols. Each operand is a four character string, with the first three characters containing the alphabetic international currency symbol in accordance with those specified in

ISO 4217, *Codes for the representation of currencies and funds*. The fourth character is the character used to separate the international currency symbol from the monetary quantity. The keyword is specified, unless the "copy" keyword is used.

int_frac_digits One or more integers separated by semicolons, representing the number of fractional digits (those to the right of the decimal delimiter) to be written in a formatted monetary quantity using "int_curr_symbol". The keyword is specified, unless the "copy" keyword is used.

int_p_cs_precedes One or more integers separated by semicolons; set to 1 if the "int_curr_symbol" precedes the value for a nonnegative formatted monetary quantity, and set to 0 if the symbol succeeds the value. If not specified, the value of "p_cs_precedes" is taken.

int_p_sep_by_space One or more integers separated by semicolons; set to 0 if no space separates the "int_curr_symbol" from the value for a nonnegative formatted monetary quantity, set to 1 if a space separates the symbol from the value, and set to 2 if a space separates the symbol and the sign string, if adjacent. If not specified, the value of "p_sep_by_space" is taken.

int_n_cs_precedes One or more integers separated by semicolons; set to 1 if the "int_curr_symbol" precedes the value for a negative formatted monetary quantity, and set to 0 if the symbol succeeds the value. If not specified, the value of "n_cs_precedes" is taken.

int_n_sep_by_space One or more integers separated by semicolons; set to 0 if no space separates the "int_curr_symbol" from the value for a negative formatted monetary quantity, set to 1 if a space separates the symbol from the value, and set to 2 if a space separates the symbol and the sign string, if adjacent. If not specified, the value of "n_sep_by_space" is taken.

int_p_sign_posn One or more integers separated by semicolons, set to a value indicating the positioning of the "positive_sign" for a nonnegative formatted monetary quantity using the "int_curr_symbol". The following integer values are defined:

0 Parentheses enclose the quantity and the "int_curr_symbol".
1 The sign string precedes the quantity and the "int_curr_symbol".
2 The sign string succeeds the quantity and the "int_curr_symbol".
3 The sign string immediately precedes the "int_curr_symbol".
4 The sign string immediately succeeds the "int_curr_symbol".

If no "int_p_sign_posn" is present the value of the "p_sign_posn" is taken.

int_n_sign_posn One or more integers separated by semicolons, set to a value indicating the positioning of the "negative_sign" for a negative formatted monetary quantity using the "int_curr_symbol". The following integer values are defined:

0 Parentheses enclose the quantity and the "int_curr_symbol".
1 The sign string precedes the quantity and the "int_curr_symbol".
2 The sign string succeeds the quantity and the "int_curr_symbol".
3 The sign string immediately precedes the "int_curr_symbol".
4 The sign string immediately succeeds the "int_curr_symbol".

If no "int_n_sign_posn" is present the value of the "n_sign_posn" is taken.

The "i18n" FDCC-set is defined as follows for the LC_MONETARY category.

```
LC_MONETARY
% This is the ISO/IEC TR 30112 i18n fdcc-set definition for
% the LC_MONETARY category.
%
int_curr_symbol     ""
currency_symbol     ""
mon_decimal_point   "<U002C>"
mon_thousands_sep   ""
mon_grouping        -1
positive_sign       ""
negative_sign       "<U002E>"
int_frac_digits     -1
frac_digits         -1
p_cs_precedes       -1
p_sep_by_space      -1
n_cs_precedes       -1
n_sep_by_space      -1
p_sign_posn         -1
n_sign_posn         -1
%
END LC_MONETARY
```

4.6 LC_NUMERIC

The LC_NUMERIC category defines the rules and symbols that are used to format nonmonetary numeric information. The operands are strings. For some keywords, the strings only can contain integers. Keywords that are not provided, string values set to the empty string (""), or integer keywords set to -1, are used to indicate that the value is unspecified. The following keywords are defined:

copy Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.

decimal_point The operand is a string containing the symbol that is

used as the decimal delimiter in numeric, nonmonetary formatted quantities. This keyword cannot be omitted and cannot be set to the empty string. In contexts where other standards limit the decimal point to a single byte, the result of specifying a multibyte operand is unspecified.

thousands_sep The operand is a string containing the symbol that is used as a separator for groups of digits to the left of the decimal delimiter in numeric, nonmonetary formatted monetary quantities. In contexts where other standards limit the "thousands_sep" to a single byte, the result of specifying a multibyte operand is unspecified.

grouping Define the size of each group of digits in formatted non-monetary quantities. The operand is a sequence of integers separated by semicolons. Each integer specifies the number of digits in each group, with the initial integer defining the size of the group immediately preceding the decimal delimiter, and the following integers defining the preceding groups. If the last integer is not -1, then the size of the previous group (if any) is repeatedly used for the remainder of the digits. If the last integer is -1, then no further grouping is performed.

The "i18n" FDCC-set is for the LC_NUMERIC category:

```
LC_NUMERIC
% This is the ISO/IEC TR 30112 i18n fdcc-set definition for
% the LC_NUMERIC category.
%
decimal_point   "<U002C>"
thousands_sep   ""
grouping        -1
%
END LC_NUMERIC
```

4.7 LC_TIME

The LC_TIME category defines the rules and symbols that are used to format date and time information.

> Note: ISO 8601 allows different formats for dates, one form is YYYY-MM-DD, another is YYYYMMDD. Each clause in this specification specifies which specific format of ISO 8601 that is used there.

The following keywords are defined:

copy Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.

abday	Define the abbreviated weekday names for calendar systems with weeks of constant length, to be referenced by the %a field descriptor. The length of the week and a Gregorian date for the first weekday is defined by the "week" keyword. The operand consists of semicolon-separated strings. The first string is the abbreviated name of the day corresponding to the first day of the week (default Sunday), the second the abbreviated name of the day corresponding to the second day of the week (default Monday), and so on.
day	Define the full weekday names for calendar systems with weeks of constant length, to be referenced by the %A field descriptor. The length of the week and a Gregorian date for the first weekday is defined by the "week" keyword. The operand consists of semicolon-separated strings. The first string is the full name of the day corresponding to the first day of the week (default Sunday), the second the full name of the day corresponding to the second day of the week (default Monday), and so on.
week	Is used to define the number of days in a week, and which weekday is the first weekday (the first weekday has the value 1), and which week is to be considered the first in a year. The first operand is an integer specifying the number of days in the week. The second operand is an integer specifying the Gregorian date in the format YYYYMMDD, and it specifies a day that is a first weekday (all other first weekdays may then be calculated by adding or subtracting a whole multiplum of the number of days in the week as specified with the first operand). The third operand is an integer specifying the weekday number to be contained in the first week of the year. The third operand may also be understood as the number of days required in a week for it to be considered the first week of the year. If the keyword is not specified the values are taken as 7, 19971130 (a Sunday), and 7 (Saturday), respectively. ISO 8601 conforming applications should use the values 7, 19971201 (a Monday), and 4 (Thursday), respectively. This keyword is optional.
abmon	Define the abbreviated month names, to be referenced by the %b field descriptor. The operand consists of twelve or thirteen semicolon-separated strings. The first string is the abbreviated name of the first month of the year (January), the second the abbreviated name of the second month, and so on.
mon	Define the full month names, to be referenced by the %B field descriptor. The operand consists of twelve or

	thirteen semicolon-separated strings. The first string is the full name of the first month of the year (January), the second the full name of the second month, and so on.
d_t_fmt	Define the appropriate date and time representation, to be referenced by the %c field descriptor. The operand consists of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain field descriptors defined in Table 3.
d_fmt	Define the appropriate date representation, to be referenced by the %x field descriptor. The operand consists of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain field descriptors defined in Table 3.
t_fmt	Define the appropriate time representation, to be referenced by the %X field descriptor. The operand consists of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain field descriptors defined in Table 3.
am_pm	Define the appropriate representation of the ante meridiem and post meridiem strings, to be referenced by the %p field descriptor. The operand consists of two strings, separated by a semicolon. The first string represents the antemeridiem designation, the last string the postmeridiem designation. The keyword is optional. If unspecified, the %p field descriptor refers to the empty string.
t_fmt_ampm	Define the appropriate time representation in the 12-hour clock format with "am_pm", to be referenced by the %r field descriptor. The operand consists of a string and can contain any combination of characters and field descriptors. If the string is empty, the 12-hour format is not supported in the FDCC-set.

The following keywords are all optional:

era	Define how years are counted and displayed for each era in a locale. The operand shall consist of semicolon-separated strings. Each string shall be an era description segment with the format: direction:offset:start_date:end_date:era_name:era_format according to the definitions below. There can be as many era description segments as are necessary to describe the different eras. NOTE: The start of an era might not be the earliest point in the era - it may be the AD 1, and increases with earlier time.
direction	Either a '+' or a '-' character. The '+' character shall indicate that years closer to the start_date have lower

	numbers than those closer to the end_date. The '-' character shall indicate that years closer to the start_date have higher numbers than those closer to the end_date.
offset	The number of the year closest to the start_date in the era, corresponding to the %Ey conversion specification
start_date	A date in the format YYYYMMDD, where YYYY, MM, and DD are the year, month, and day numbers respectively according to ISO 8601 of the start of the era. Years prior to AD 1 shall be represented as negative numbers.
end_date	The ending date of the era, in the same format as the start_date, or one of the two special values "-*" or "+*". The value "-*" shall indicate that the ending date is the beginning of time. The value "+*" shall indicate that the ending date is the end of time.
era_name	A string representing the name of the era, corresponding to the %EC conversion specification.
era_format	A string for formatting the year in the era, corresponding to the %EY conversion specification.
era_year	Define the format of the year in alternate Era format, corresponding to the %EY field descriptor.
era_d_t_fmt	Define the format of the date and time in alternate Era notation, corresponding to the %Ec field descriptor.
era_d_fmt	Define the format of the date in alternate Era notation, corresponding to the %Ex field descriptor.
era_t_fmt	Define the format of the time in alternate Era notation, corresponding to the %EX field descriptor.
alt_digits	Define alternate symbols for digits, corresponding to the %O field descriptor modifier. The operand consists of semicolon-separated strings. The first string is the alternate symbol corresponding with zero, the second string the symbol corresponding with one, and so on. Up to 100 alternate symbol strings can be specified. The %O modifier indicates that the string corresponding to the value specified via the field descriptor is used instead of the value.
first_weekday	Define the first day to be displayed, for example in a calendar display utility. The operand is an integer specifying the day number (1 = first) according to the information specified with the "day" keyword. The keyword may be omitted, and then the value 1 is taken, corresponding to Sunday for a week beginning Sunday, or to Monday for a week beginning Monday.
first_workday	Define the first workday as an integer according to the day numbering specified with the "week" keyword.
cal_direction	Define the direction of the display of dates, for example in a calendar display utility. The operand is an integer, and the following values are defined:

1 left-right from top
2 top-down from left
3 right-left from top

The keyword may be omitted, and then the value 1 is taken.

timezone Define one or more timezones, each defined by a string, and the strings separated by a <semicolon>. In the following the characters <, >, [and] are used as metacharacters. Only characters with a visible glyph from the portable character set may be used, except in the <std> and <dst> fields. The syntax of a string is:

<std><offset><dst>[<offset>][,<rule>[,<rule>...]];

where

<std> and <dst> Indicates no less than three, nor more than 10 characters that are the designation for the standard <std>, or Daylight Savings Time or summer time <dst> zone. Only <std> is required; if <dst> is missing, then Daylight Savings Time or summer time does not apply in this category. Upper- and lowercase letters are explicitly allowed. Any characters except a leading colon <:> or digits, the comma <,>, the minus <->, the plus <+>, and the null character are permitted to appear in these fields, but their meaning is unspecified.

<offset> Indicates the value one must add to the local time to arrive at the Coordinated Universal Time. The <offset> has the form:

hh[:mm[:ss]]

The minutes (mm) and seconds (ss) are optional. The hour (hh) is required and may be a single digit. The <offset> following <std> is required. If no <offset> follows <dst>, summer time is assumed to be one hour ahead of standard time. One or more digits may be used; the value is always interpreted as a decimal number. The hour is between zero and 24, and the minutes (and seconds) - if present - is between zero and 59. If preceded by a "-", the time zone is east of the Prime Meridian; otherwise it is west of (which may be indicated by an optional preceding "+").

<rule> A specification for Daylight Savings Time changes that indicates when to change to and back from summer time. The <rule> has the form:

<date>[/<time>/<year>],<date>[/<time>

/<year>]

where the first <date> describes when the change from standard time to summer time occurs, and the second <date> describes when the change back happens. Each <time> field describes when, in current local time, the change to the other time is made. The first <year> field defines the beginning of the validity of this rule, and the second <year> field defines the end of the validity of the rule. A number of rules may be given.

The format of <date> is one of the following:

J<n> The Julian day <n> (1 <= n <= 365) Leap years are not counted. That is, in all years - including leap years - February 28 is day 59 and March 1 is day 60. It is impossible to explicitly refer to the occasional February 29.
<n> The zero-based Julian day (0 <= n <= 365). Leap years are counted and it is possible to refer to February 29.
M<m>.<n>.<d>
the <d>th day (0 <= d <= 7) of week <n> of month <m> (1 <= n <= 5, 1 <= m <= 12, where week 5 means "the last <d> day in month <m>" which may occur in either the fourth or fifth week). Week 1 is the first week in which the <d>th day occurs. Day zero and day seven is Sunday.

The <time> has the same format as <offset> except that no leading sign ("-" or "+") is allowed. The default, if <time> is not given, is "02:00:00".

The <year> has the format YYYY.

NOTE: This way of specifying the timezone is compatible with the format for the environment variable TZ described in the POSIX standard.

4.7.1 Date Field Descriptors

The LC_TIME category defines the interpretation of a number of field descriptors. The field descriptors are also available in the definitions with the following LC_TIME keywords: "d_t_fmt", "d_fmt", "t_fmt", "t_fmt_ampm", "era", "era_d_t_fmt", "era_d_fmt", and "era_t_fmt". A field descriptor may not be used with the LC_TIME keywords defining it.

Table 3: Field descriptors for the date field

%a	FDCC-set's abbreviated weekday name.
%A	FDCC-set's full weekday name.
%b	FDCC-set's abbreviated month name.
%B	FDCC-set's full month name.
%c	FDCC-set's appropriate date and time representation.
%C	Century (a year divided by 100 and truncated to integer) as decimal number (00-99).
%d	Day of the month as a decimal number (01-31).
%D	Date in the format mm/dd/yy.
%e	Day of the month as a decimal number (1-31 in at two-digit field with leading <space> fill).
%F	The date in the format YYYY-MM-DD (An ISO 8601 format).
%g	Week-based year within century, as a decimal number (00-99).
%G	Week-based year with century, as a decimal number (for example 1997).
%h	A synonym for %b.
%H	Hour (24-hour clock), as a decimal number (00-23).
%I	Hour (12-hour clock), as a decimal number (01-12).
%j	Day of the year, as a decimal number (001-366).
%m	Month, as a decimal number (01-13).
%M	Minute, as a decimal number (00-59).
%n	A <newline> character.
%p	FDCC-set's equivalent of either AM or PM.
%r	12-hour clock time (01-12), using the AM/PM notation.
%R	24-hour clock time, in the format "%H:%M".
%S	Seconds, as a decimal number (00-61).
%t	A <tab> character.
%T	24-hour clock time, in the format HH:MM:SS.
%u	Weekday, as a decimal number (1(Monday)-7).
%U	Week number of the year (Sunday as the first day of the week) as a decimal number (00-53). All days in a new year preceding the first Sunday are considered to be in week 0.
%v	Week number of the year, as a decimal number with two digits including a possible leading zero, according to "week" keyword.
%V	Week of the year (Monday as the first day of the week), as a decimal number (01-53). The method for determining the week number is as specified by ISO 8601.
%w	Weekday, as a decimal number (0(Sunday)-6).

%W	Week number of the year (Monday as the first day of the week), as a decimal number (00-53). All days in a new year preceding the first Monday are considered to be in week 0.
%x	FDCC-set's appropriate date representation.
%X	FDCC-set's appropriate time representation.
%y	Year within century (00-99).
%Y	Year with century, as a decimal number.
%z	The offset from UTC in the ISO 8601 format "-0430" (meaning 4 hours 30 minutes behind UTC, west of Greenwich), or by no characters if no time zone is determinable.
%Z	Time-zone name, or no characters if no time zone is determinable.
%%	A <percent-sign> character.

NOTE: %g, %G and %V give values according to the ISO 8601 week-based year. In this system, weeks begin on a Monday and week 1 of the year is the week that includes 4th January, which is also the week that includes the first Thursday of the year, and is also the first week that contains at least four days in the year. If the first Monday of the year is the 2nd, 3rd or 4th, the preceding days are part of the last week of the preceding year; thus, for Saturday 2nd January 1999, %G is replaced by 1998 and %V is replaced by 53. If the 29th, 30th or 31st December is a Monday, it and any following days are part of week 1 of the following year. Thus, for Tuesday 30th December 1997, %G is replaced by 1998 and %V is replaced by 1.

4.7.2 Modified Field Descriptors

Some field descriptors can be modified by the E and O modifier characters to indicate a different format or specification as specified in the LC_TIME FDCC-set description. If the corresponding keyword (see "era", "era_year", "era_d_t_fmt", "era_d_fmt", "era_t_fmt" and "alt_digits") is not specified for the current FDCC-set, the unmodified field descriptor value is used.

%Ec	FDCC-set's alternate date and time representation.
%EC	The name of the base year (period) in the FDCC-set's alternate representation.
%Ex	FDCC-set's alternate date representation.
%EX	FDCC-set's alternate time representation.
%Ey	Offset from %EC (year only) in the FDCC-set's alternate representation.
%EY	Full alternate year representation.
%Od	Day of month using the FDCC-set's alternate numeric symbols.
%Oe	Day of month using the FDCC-set's alternate numeric symbols.
%Of	Weekday as a decimal number according to alt_day (1 is first day).
%OH	Hour (24-hour clock) using the FDCC-set's alternate numeric symbols.
%OI	Hour (12-hour clock) using the FDCC-set's alternate numeric symbols.
%Om	Month using the FDCC-set's alternate numeric symbols.
%OM	Minutes using the FDCC-set's alternate numeric symbols.
%OS	Seconds using the FDCC-set's alternate numeric symbols.

%Ou	Weekday as a number in the alternate representation of the FDCC-set (Monday=1).
%OU	Week number of the year (Sunday as the first day of the week) using the FDCC-set's alternate numeric symbols.
%OV	Week number of the year (Monday as the first day of the week, ISO 8601 rules) using the alternate numeric symbols of the FDCC-set.
%Ow	Weekday as number in the FDCC-set's alternate representation (Sunday=0).
%OW	Week number of the year (Monday as the first day of the week) using the FDCC-set's alternate numeric symbols.
%Oy	Year (offset from %C) in alternate representation.

4.7.3 "i18n" LC_TIME category

The "i18n" LC_TIME category is (following ISO 8601):

```
LC_TIME
% This is the ISO/IEC TR 30112 "i18n" definition for
% the LC_TIME category.
%
% Weekday and week numbering according to ISO 8601
abday   "<U0031>";"<U0032>";"<U0033>";"<U0034>";/
        "<U0035>";"<U0036>";"<U0037>"
day     "<U0031>";"<U0032>";"<U0033>";"<U0034>";/
        "<U0035>";"<U0036>";"<U0037>"
week    7;19971201;4
abmon   "<U0030><U0031>";"<U0030><U0032>";"<U0030><U0033>";/
        "<U0030><U0034>";"<U0030><U0035>";"<U0030><U0036>";/
        "<U0030><U0037>";"<U0030><U0038>";"<U0030><U0039>";/
        "<U0031><U0030>";"<U0031><U0031>";"<U0031><U0032>"
mon     "<U0030><U0031>";"<U0030><U0032>";"<U0030><U0033>";/
        "<U0030><U0034>";"<U0030><U0035>";"<U0030><U0036>";/
        "<U0030><U0037>";"<U0030><U0038>";"<U0030><U0039>";/
        "<U0031><U0030>";"<U0031><U0031>";"<U0031><U0032>"
am_pm   "";""
% Date formats following ISO 8601
% Appropriate date and time representation (%c)
%       "%F %T"
d_t_fmt "<U0025><U0046><U0020><U0025><U0054>"
%
% Appropriate date representation (%x)   "%F"
d_fmt   "<U0025><U0046>"
%
% Appropriate time representation (%X)   "%T"
t_fmt   "<U0025><U0054>"
t_fmt_ampm ""
%
END LC_TIME
```

4.8 LC_MESSAGES

The LC_MESSAGES category defines the format and values for affirmative and negative responses. The operands are strings or extended regular expressions to specify which response strings that should be considered matches; see ISO/IEC 9945-1:2003 clause 9.3 for a definition of extended regular

expressions. The following keywords are defined:

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.
yesexpr	The operand consists of an extended regular expression that describes the acceptable affirmative response to a question expecting an affirmative or negative response.
noexpr	The operand consists of an extended regular expression that describes the acceptable negative response to a question expecting an affirmative or negative response.
yesstr	The operand consists of a string that describes the affirmative response to a question.
nostr	The operand consists of a string that describes the negative response to a question.

The "i18n" LC_MESSAGES category is:

```
LC_MESSAGES
% This is the ISO/IEC TR 30112 "i18n" definition for
% the LC_MESSAGES category.
%
yesexpr "<U005B><U002B><U0031><U005D>"
noexpr  "<U005B><U002D><U0030><U005D>"
END LC_MESSAGES
```

Note: This uses regular expression syntax with brackets ([]) to for example specify that both <+> and <1> is allowed as an affirmative answer.

4.9 LC_XLITERATE

The LC_XLITERATE category defines formats to transform strings, by transforming substrings in the source to substrings in the target string. The target is the culture of the FDCC-set in question. The capabilities can be used for simple transliteration or fallback based on substring substitution, while more advanced transliteration schemes, for example based on pattern matching, sound equivalences, or using a database, is either cumbersome to specify, or not addressed. The transliteration may for example be from the Cyrillic script to the Latin script.

Note: transliteration is for backwards compatibility reasons also possible with the LC_CTYPE category, see clause 4.3.2.

Transliteration of an incoming character string to a character string in a FDCC-set can be specified with the following transliteration keywords and transliteration statements.

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.

include	The name of the FDCC-set in text form to transliterate from, and the repertoiremap for the FDCC-set to be used for the definition of the transliteration statements. Other transliteration statements may follow to replace specification of the copied FDCC-set. This keyword is optional.
default_missing	defines a string of one or more characters to be put in the output string if no transliteration statement can be applied to a input <transliteration-source>. This keyword is optional.
translit_ignore	defines a set of characters, separated by semicolons, that are to be ignored in the incoming character string, that is, each of the occurrences of such characters is treated as the empty string. The characters may use the notations defined in 4.3 for lists of characters. This keyword is optional.
redefine	This keyword introduces a list of transliteration statements where each of the <transliteration_source> strings have been defined previously in the specification, and the new transliteration statements then replaces the old transliteration statements for the <transliteration_source> strings specified. This keyword is optional.

4.9.1 Transliteration statements

The syntax for a transliteration statement is:

"%s %s;%s;...;%s\n",<transliteration_source>,<transliteration_string>,...

Each <transliteration_source> consists of one or more characters (in any of the forms defined in 4.1.1). The <transliteration_source> that is the longest in terms of number of characters that match the input string is the one selected for transliteration.

If a transliteration statement contains more than one <transliteration_string>, the order that each <transliteration_string> occurs in the transliteration statement defines the precedence order for choosing a particular <transliteration_string> to substitute for the <transliteration_source>. When a process makes use of a transliteration statement to transliterate text, and that transliteration statement contains more than one <transliteration_string>, that process chooses the first <transliteration_string>, in the defined precedence order, that satisfies the requirements of the transliteration.

Note: the exact definition of the concept of satisfying the requirements of the transliteration is outside the context of this International Standard. If, for example, a transliteration involves a

change in the coded character set of a string, a <transliteration_string> must be chosen, all of whose elements are members of that coded character set. In order to determine this, it would be expected that a repertoire describing which characters are to be present in the resulting transformed string be available to the transliteration API. Also, a transliteration may involve requirements such as that string length not change under transliteration. Such requirements may also affect the choice among alternative <transliteration_string> values.

If more than one transliteration statement is given for a given <transliteration_source> this is an error, and duplicate transliteration statements are ignored. Tailoring of transliteration statements may be done via the "redefine" keyword.

4.9.2 "include" keyword

The "include" keyword specifies a set of transliteration statements in text form to be included in the applied transliteration. The syntax of the "include" statement is:

"include %s;%s\n", <FDCC-set>, <repertoiremap>

<FDCC-set> is a string identifying the FDCC-set to be included from.

<repertoiremap> is a string identifying the repertoiremap used in the FDCC-set being included, and is used to map character specifications from the specified FDCC-set into the current FDCC-set.

4.9.3 Example of use of transliteration

```
LC_XLITERATE
include "de_DE";"de_repmap"
default_missing <?>
translit_ignore <U3200>..<UFAFF>
<ae>       <a:>;<e*>;"<a><e>";"<e>"
<s>        <s*>;<s=>
"<K><O>"   <KO>
END LC_XLITERATE
```

The "LC_XLITERATE" statement introduces the transliteration category.

The "include" keyword specifies that the FDCC-set "de_DE" is copied and that the repertoiremap "de_repmap" is used to define the symbolic character names in the FDCC-set "de_DE".

The "default_missing" keyword introduces the character sequence "<?>" as the string to transform into for input characters that cannot be transformed into other strings, because no transliteration statement is applicable to the character.

The "translit_ignore" keyword specifies that a set of Ideographic characters, Hangul, East Asian symbols and the private use area etc. (the range <U3200>..<UFAFF>) is ignored for the transliteration.

The next 3 lines are transliteration statements.

The first transliteration statement defines a number of transliterations for the LATIN LETTER AE, including into LATIN LETTER A WITH DIAERESIS, GREEK LETTER EPSILON, the two Latin letters A and E, and finally the LATIN LETTER E.

The second transliteration statement defines transliteration of the LATIN LETTER S into GREEK LETTER

SIGMA, and CYRILLIC LETTER ES.

The third transliteration statement transliterates the two Latin letters K and O into the Japanese Hiragana character KO.

The transliteration category is terminated via the "END LC_XLITERATE" statement in the above example.

There is no "i18n" entry for the LC_XLITERATE category

4.10 LC_NAME

The LC_NAME category defines formats to be used in addressing a person, e.g. in a postal address or in a letter. The following keywords are defined:

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.
name_fmt	Define the appropriate representation of a person's name and title. The operand consists of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain field descriptors defined below.
name_gen	The operand is a string defining a salutation valid for all persons.
name_miss	The operand is a string defining a salutation valid for unmarried females.
name_mr	The operand is a string defining a salutation valid for males.
name_mrs	The operand is a string defining a salutation valid for married females.
name_ms	The operand is a string defining a salutation valid for all females.

NOTE: There are a number of variations for addressing a person among the cultures. Middle names are not used in many countries and even the family name is not used in some countries. In other countries there is extensive use of one or more middle names and corresponding initials. The specification below should be regarded as a starting point for this problem.

The LC_NAME category defines the interpretation of a number of field descriptors. The field descriptors are also available in the definitions with the following LC_NAME keywords: "name_fmt".

Field descriptors for the "name_fmt" keyword:

%f	Family names.
%F	Family names in uppercase.
%g	First given name.
%G	First given initial.
%l	First given name with Latin letters. In some cultures, eg on Taiwan it is customary to also have a first name written with Latin letters,

	although the rest of the name is written in another script.
%o	Other shorter name, eg. "Bill".
%m	Additional given names.
%M	Initials for additional given names.
%p	Profession.
%s	Salutation, such as "Doctor"
%S	Abbreviated salutation, such as "Mr." or "Dr."
%d	Salutation, using the FDCC-sets conventions, with 1 for the name_gen, 2 for name_mr, 3 for name_mrs, 4 for name_miss, 5 for name_ms.
%t	If the preceding field descriptor resulted in an empty string, then the empty string, else a <space>.

Each field descriptor may have an <R> after the <%> to specify that the information is taken from a Romanized version string of the entity. An initial is any string, normally consisting of one letter and a punctuation mark; the Dutch "IJ" is an example of a two character initial.

The "i18n" LC_NAME category is:

```
LC_NAME
% This is the ISO/IEC TR 30112 "i18n" definition for
% the LC_NAME category.
name_fmt    "<U0025><U0070><U0025><U0074><U0025><U0067><U0025><U0074>/
<U0025><U006D><U0025><U0074><U0025><U0066>"
% This corresponds to "%p%t%g%m%t%f" which is
% Profession Primary Additionals Family
END LC_NAME
```

4.11 LC_ADDRESS

The LC_ADDRESS category defines formats to be used in specifying a location like a person's home or office, for use in a postal address or in a letter, and other items related to geography, including natural language. All keywords are strings and may contain non-digits, and all keywords are optional. The following keywords are recognized:

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.
postal_fmt	Define the appropriate representation of a postal address such as street and city. The proper formatting of a person's name and title is done with the "name_fmt" keyword of the LC_NAME category. The operand consists of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain field descriptors defined below.

country_name	The operand is a string with the name of the country in the language of the FDCC-set.
country_post	The operand is a string with the abbreviation of the country, used for postal addresses, for example by the CEPT-MAILCODE codes designating countries in Europe. Other abbreviation systems are also allowed, and there is no specific way to identify which abbreviation system is being used.
country_isbn	The operand is a string with one or more numbers separated by a semicolon that represent the ISBN numbers allocated to the country.
lang_name	The operand is a string with the name of the language in the language of the FDCC-set.
lang_ab2	The operand is a string with the two-letter abbreviation of the language, according to ISO 639.
lang_ab3_term	The operand is a string with the three-letter abbreviation of the language for terminology use, according to ISO 639-2.
lang_ab3_lib	The operand is a string with the three-letter abbreviation of the language for library use, according to ISO 639-2. If not specified, the value of the "lang_ab3_term" keyword is taken.

Note: The "lang_ab3_term" and "lang_ab3_lib" keywords will in most cases contain the same value, but they may differ, e.g. the values for the German language is "deu" and "ger" respectively.

The LC_ADDRESS category defines the interpretation of a number of field descriptors. The field descriptors are also available in the definitions with the following LC_ADDRESS keywords: "postal_fmt".

Field descriptors for the "postal_fmt" keyword:

%n	Person's name, possibly constructed with the LC_NAME "name_fmt" keyword.
%a	Care of person, or organization.
%f	Firm name.
%d	Department name.
%b	Building name.
%s	Street or block (eg. Japanese) name.
%h	House number or designation.
%N	Insert an <end-of-line> if the previous descriptor's value was not an empty string; otherwise ignore.
%t	Insert a <space> if the previous descriptor's value was not an empty string; otherwise ignore.
%r	Room number, door designation.
%e	Floor number.

%C	Country designation, from the <country_post> keyword.
%l	Local township within town or city
%z	Zip number, postal code.
%T	Town, city.
%S	State, province, or prefecture.
%c	Country, as taken from data record.

Each field descriptor may have an <R> after the <%> to specify that the information is taken from a Romanized version string of the entity.

NOTE: There are a number of variations for specifying a location among the cultures. Some of the information, like the middle names, or even the family name, is not used in some cultures. The specification here should be regarded as a starting point for this problem.

Examples:

A specification for the USA could be:

"%n%N%a%N%d%N%f%N%b%N%h %s%N%e %r%N%l%N%C-%z %T%, %S %z%N%c%N"

Giving:

Person's name
C/o address
Department
Firm
Building
number street
floor room
Local Town
City, State Zip
Country

An example for South Korea could be:
"%S %T %l %s %h %N%f %d%N%b %e %r%N%n %a%N%z"

Giving:
State City Town Street number
Firm department
Building floor room
Person's name C/o address
Zip

The "i18n" LC_ADDRESS category is:

```
 LC_ADDRESS
 % This is the ISO/IEC TR 30112 "i18n" definition for
 % the LC_ADDRESS category.
 %
 postal_fmt    "<U0025><U006E><U0025><U004E>/
 <U0025><U0061><U0025><U004E><U0025><U0066><U0025><U004E>/
 <U0025><U0064><U0025><U004E><U0025><U0062><U0025><U004E><U0025><U0073>/
 <U0020><U0025><U0068><U0020><U0025><U0065><U0020><U0025><U0072>/
 <U0025><U004E><U0025><U006C><U0025><U004E><U0025><U0043><U002D>/
 <U0025><U007A><U0020><U0025><U0054><U0025><U004E>/
 <U0025><U0053><U0025><U004E><U0025><U0063><U0025><U004E>"
 %
 % "%n%N%a%N%f%N%d%N%b%N%s %h %e %r%N%l%N%C-%z %T%N%S%N%c%N"  resulting in
 %  Person's_Name
 %  C/o_person_or_org
 %  Firm
 %  Department
```



```
%  Building_name
%  Street_or_block number floor room
%  Local_township
%  Country-Zip City
%  State_or_province
%  Country
%
END LC_ADDRESS
```

4.12 LC_TELEPHONE

The LC_TELEPHONE category defines formats to be used with telephone services. All keywords are optional. The strings are not restricted in what characters they can contain. The following keywords are defined:

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.
tel_int_fmt	Define the appropriate representation of a telephone number for international use. The operand consists of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain field descriptors defined below.
tel_dom_fmt	Define the appropriate representation of a telephone number for domestic use. The operand consists of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain field descriptors defined below.
int_select	The operand is a string with the digits used to call international telephone numbers.
int_prefix	The operand is a string with the prefix used from other countries to call the area.

The LC_TELEPHONE category defines the interpretation of a number of field descriptors. The field descriptors are also available in the definitions with the following LC_TELEPHONE keywords: "tel_int_fmt" and "tel_dom_fmt".

%a	area code without nationwide prefix (prefix is often <0>).
%A	area code including nationwide prefix (prefix is often <0>).
%l	local number (within area code).
%e	extension (to local number)
%c	country code
%C	alternate carrier service code used for dialling abroad
%t	Insert a <space> if the previous descriptor's value was not an empty string; otherwise ignore.

The "i18n" LC_TELEPHONE category is:

```
LC_TELEPHONE
% This is the ISO/IEC TR 30112 "i18n" definition for
% the LC_TELEPHONE category.
```



```
%
tel_int_fmt    "<U002B><U0025><U0063><U0020><U0025><U0061><U0025><U0074>/
<U0025><U006C>"
% "+%c %a%t%l" which is
%  +country area local
END LC_TELEPHONE
```

4.13 LC_PAPER

The LC_PAPER category defines the paper size. The following keywords shall be defined:

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword shall be specified.
height	Shall be used to specify the height of the paper. The operand is an integer and the value is the height measured in millimetres.
width	Shall be used to specify the width of the paper. The operand is an integer and the value is the width measured in millimetres.

Note: If the height is greater than the width, it is called to be in portrait position, else it is called to be in landscape position.

The "i18n" LC_PAPER category is:

```
LC_PAPER
% This is the ISO/IEC TR 30112 "i18n" definition for
% the LC_PAPER category.
%
height   297
width    210
END LC_PAPER
```

4.14 LC_MEASUREMENT

The LC_MEASUREMENT category defines which measurement system in use. All keywords are optional. The following keywords shall be defined:

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword shall be specified.
measurement	Shall be used to define the measurement system in use. The operand is an integer. The following values are defined: 1 ISO 1000 2 U.S.A. measurement 3 other

The "i18n" LC_MEASUREMENT category is:

```
LC_MEASUREMENT
% This is the ISO/IEC TR 30112 "i18n" definition for
% the LC_MEASUREMENT category.
```



```
%
measurement    1
END LC_MEASUREMENT
```

4.15 LC_KEYBOARD

The LC_KEYBOARD category defines the possible keyboards. The following keywords shall be defined:

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword shall be specified.
keyboards	Shall be used to specify possible keyboards to be used with this FDCC-set, the first being the one chosen if none is explicitly chosen. The operand is one or more stings separated by semicolons, each string identifying a keyboard in an implementation defined way.

The "i18n" LC_KEYBOARD category is:

```
LC_PAPER
% This is the ISO/IEC TR 30112 "i18n" definition for
% the LC_KEYBOARD category.
%
keyboards   “iso/iec-9995”
END LC_KEYBOARD
```

5. CHARMAP

A character set description may exist for each coded character set supported by the implementation. This file is referred to elsewhere in this International Standard as a charmap.

A conforming charmap to be used with a FDCC-set supports the portable character set specified in Table 1 of clause 3.2.3.

Conforming charmaps specify certain character and character set attributes, as defined in 5.1.

5.1 Character Set Description Text

The character set description text (charmap) describes the mapping between symbolic character names and actual encoding of a coded character set. It is used to bind the symbolic character names in a FDCC-set to an actual encoding, so an application can process data in this encoding.

The following declarations can precede the character definitions. Each consist of the symbol shown in the following list, starting in column 1, including the surrounding brackets, followed by one of more "blank"s, followed by the value to be assigned to the symbol. If any of the declarations are included, they are

specified in the order shown in the following list:

<**code_set_name**> The name of the coded character set for which the character set description text is defined. The characters of the name are taken from the set of characters with visible glyphs defined in Table 1 of clause 3.2.3.

<**mb_cur_max**> The maximum number of bytes in a multibyte character. This defaults to 1.

<**mb_cur_min**> An unsigned positive integer value that defines the minimum number of bytes in a character for the encoded character set. The value is less or equal to "mb_cur_max". If not specified, the minimum number is equal to "mb_cur_max".

<**escape_char**> The escape character used to indicate that the characters following is interpreted in a special way, as defined later in this subclause. This defaults to backslash (\). The character slash (/) is used in all the following text and examples, unless otherwise noted.

<**comment_char**> The character that when placed in column 1 of a charmap line, is used to indicate that the line is ignored. The default character is the number sign (#). The character percent-sign (%) is used in all the following text and examples, unless otherwise noted.

<**repertoiremap**> The name of the repertoiremap used to define the symbolic character names in the charmap. The characters of the name are taken from the set of characters with visible glyphs defined in Table 1 of clause 3.2.3.

<**escseq2022**> defines the escape sequences for ISO 2022 shifting for the coded character set defined by the charmap. The semicolon-separated operands are all strings with characters taken from the set of characters with visible glyphs defined in table 1. The first operand defines the g-set or c-set to be defined, and the following values are defined: c0, c1, g0, g1, g2, g3. The second operand defines what range of characters in the charmap is affected, and the values defined are: c0, c1, g0, g1. The third operand is the escape sequence that is defined.

<**addset**> the name of the charmap to be added to the current coded character set, and to be selected by the escape sequences defined by <escseq2022> of the added

	charmap.
<**include**>	include the encoding of another charmap in the current charmap. The semicolon-separated operands are all strings with characters taken from the set of characters with visible glyphs defined in table 1. The first operand defines the g-set or c-set to be defined in the current charmap, and the following values are defined: c0, c1, g0, g1, g2, g3. The second operand defines a range of characters in the referenced charmap, and the values defined are: c0, c1, g0, g1. The third operand is the name of the charmap to be included. The coded character sets are defined initially for the encoding, and therefore do not need escape sequences for identification. If two g0 sets are defined, the second is switched to using the SHIFT OUT control character, while the first is shifted to using the SHIFT IN control character.

The character set mapping definitions are all the lines immediately following an identifier line containing the string "CHARMAP" starting in column 1, and preceding a trailer line containing the string "END CHARMAP" starting in column 1. Empty lines and lines containing a <comment_char> in the first column are ignored. Each non-comment line of the character set mapping definition (i.e., between the "CHARMAP" and "END CHARMAP" lines of the text) is in one of the following syntaxes.

"%s %s %s\n", <symbolic-name>,<encoding>,<comments>

"%s...%s %s %s\n", <symbolic-name>,<symbolic-name>,<encoding>,<comments>

"%s....%s %s %s\n", <symbolic-name>,<symbolic-name>,<encoding>,<comments>

"%s..%s %s %s\n", <symbolic-name>,<symbolic-name>,<encoding>,<comments>

In the first syntax, the line of the character set mapping definition starts with the symbolic name, immediately preceded by a <less-than> character and immediately followed by a <greater-than> character. Symbolic names only contain characters from the set shown with a visible glyph in Table 1 of clause 3.2.3.

The same symbolic name may occur several times, with different values. The first value is the one used when generating an encoding, while the other values are accepted in decoding. Symbolic names may be included to identify values

that can overlap with each other or with the values of the symbolic names shown in Table 1 of clause 3.2.3. It is possible to specify symbolic names for which no encoding exists in the encoded character set, by not specifying a value.

In the second and third syntax (symbolic decimal ellipsis), the line in the character set mapping defines a range of one or more symbolic names. The difference between the second and the third syntax is the number of dots in the ellipsis: the second has 3 dots, the third has 4 dots. In these forms the symbolic names consist of zero or more non-numeric characters from the set shown with visible glyphs in Table 1 of clause 3.2.3, followed by an integer formed by one or more decimal digits. The characters preceding the integer are identical in the two symbolic names, and the integer formed by the digits in the second symbolic name are identical to or greater than the integer formed by the digits in the first name. This is interpreted as a series of symbolic names formed from the common part and each of the integers in decimal format between the first and the second integer, inclusive, and with a length of the symbolic names generated that is equal to the length of the first (and also the second) symbolic name. As an example, <j0101>....<j0104> is interpreted as the symbolic names <j0101>, <j0102>, <j0103>, and <j0104>, in that order.

Note: The rationale to allow both a 3-dot and a 4-dot symbol for symbolic decimal ellipses is that in the POSIX standard the decimal symbolic ellipses was defined by a 3-dot symbol for charmaps, while the 3-dot symbol was an absolute ellipses for POSIX locales, and this International Standard specifies a 4-dot symbol for the decimal symbolic ellipses. The 3-dot symbolic decimal ellipses in charmaps is deprecated.

In the fourth syntax (symbolic hexadecimal ellipsis, with two dots), the line in the character set mapping defines a range of one or more symbolic names. In this form the symbolic names consist of zero or more non-numeric characters from the set shown with visible glyphs in Table 1 of clause 3.2.3, followed by an integer formed by one or more hexadecimal digits, using uppercase letters only for the range "A" to "F". The characters preceding the hexadecimal integer are identical in the two symbolic names, and the integer formed by the hexadecimal digits in the second symbolic name is identical to or greater than the integer formed by the hexadecimal digits in the first name. This is interpreted as a series of symbolic names formed from the common part and each of the integers in hexadecimal format using uppercase letters only between the first and the second integer, inclusive, and with a length of the symbolic names generated that is equal to the length of the first (and also the second) symbolic name. As an example, <U010E>..<U0111> is interpreted as the symbolic names <U010E>, <U010F>, <U0110>, and <U0111>, in that order.

The encoding part is expressed as one (for single-byte values) or more concatenated decimal, octal or hexadecimal constants (hexadecimal constants are recommended). Decimal constants are represented by two or three decimal digits, preceded by the escape character and the lowercase letter "d"; for

example /d05, /d97, or /d143. Hexadecimal constants are represented by two hexadecimal digits, preceded by the escape character and the lowercase letter "x"; for example /x05, /x61, or /x8f. Octal constants are represented by two or three octal digits, preceded by the escape character; for example /05, /141, or /217. In a charmap, each constant should represent an 8 bit byte for portability reasons. Applications supporting other byte sizes may allow constants to represent values larger than those that can be represented in 8 bit bytes, and to allow additional digits in constants. When constants are concatenated for multibyte character values, they may be of different types, and interpreted in byte order from the first to the last with the least significant byte of the multibyte character specified by the last byte. The manner in which these constants are represented in the character stored in the system is application defined. Omitting bytes from a multibyte character produces undefined results.

In lines defining ranges of symbolic names, the encoded value is the value for the first symbolic name in the range (the symbolic name preceding the ellipsis). Subsequent symbolic names defined by the range have encoding values in increasing order. For example the line

<j0101>....<j0104> /d129/d254

is interpreted as

<j0101> /d129/d254
<j0102> /d129/d255
<j0103> /d130/d000
<j0104> /d130/d001

The comments parameter is optional.

Example of using ISO 2022 techniques:

The following example defines two coded character sets, a 7-bit and a 14-bit. They are then merged into one encoding. It is an example on how encodings used in Eastern Asia could be specified.

The 7-bit charmap

```
<escape_char> /
<comment_char> %
% The 7-bit charmap defines both control and graphic characters
<code_set_name>  "eastern7bit"
<escseq2022>         "c0";"c0","/x21/x40"
<escseq2022>         "g0";"g0","/x28/x48"
<escseq2022>         "g1";"g0","/x29/x48"
<escseq2022>         "g2";"g0","/x2A/x48"
<escseq2022>         "g3";"g0","/x2B/x48"

CHARMAP
<tab>            /x08
<newline>        /x0D
<a>              /x61
% more character encodings to be defined here
END CHARMAP
```


The 14-bit charmap

```
<escape_char> /
<comment_char> %
<code_set_name>   "eastern14bit"
<mb_cur_max>      2
<esqseq2022>      "g0";"g0";"/x24/x40"
<esqseq2022>      "g1";"g0";"/x24/x29/x40"
<esqseq2022>      "g2";"g0";"/x24/x2A/x40"
<esqseq2022>      "g3";"g0";"/x24/x2B/x40"
CHARMAP
<U0165>           /d036/d055   % the character codes are only examples
<U0166>           /d036/d056
% more character encodings to be defined here
END CHARMAP
```

The merged encoding

```
<escape_char> /
<comment_char> %
<code_set_name>    "shift-eastern"
<mb_cur_max>       2
<mb_cur_min>       1
<include>          "c0";"c0";"eastern7bit"
<include>          "g0";"g0";"eastern7bit"
<include>          "g1";"g0";"eastern14bit"
% This defines the g0 values of "eastern14bit" (without the 8th
% bit set) to be the g1 in this encoding (with the 8th bit set).
%
% So the bytes without the 8th bit set is from the "eastern7bit"
% coded character set, while bytes with the 8th bit set are from
% the 14-bit set.
```

Another merged encoding using the same charmaps:

```
<escape_char> /
<comment_char> %
<code_set_name>    "EUC-eastern"
<mb_cur_max>       2
<mb_cur_min>       1
<include>          "c0";"c0";"eastern7bit"
<include>          "g0";"g0";"eastern7bit"
<include>          "g0";"g0";"eastern14bit"
% As there are two "g0" sets defined, the first referenced is the
% initial g0 set, while the second can be shifted to via the SHIFT OUT
% control character. The first can then be shifted to by the SHIFT IN
% control character.
```

WIDTH section

After the "END CHARMAP" statement the following declarations may follow. Each consists of the keyword shown in the following list, starting in column 1, followed by the value(s) to be associated to the keyword, as defined below.

WIDTH An unsigned positive integer value defining the column width for the characters in the coded character set. Coded character values are defined using symbolic character names followed by a column width value. Defining a character with more than one WIDTH produces undefined results. The END WIDTH keyword is used to terminate the WIDTH definitions.

Note: The WIDTH section is mostly intended to cover halfwidth and fullwidth characters in

fixed width character sets of Eastern Asia and corresponding characters in ISO/IEC 10646.

WIDTH_DEFAULT An unsigned positive integer value defining the column width for any character not listed by one of the WIDTH keywords. If no WIDTH_DEFAULT keyword is included in the charmap, the default character width is 1.

Example:

After the "END CHARMAP" statement, a syntax for width definition would be:

```
WIDTH
<A> 1
<B> 1
<j0101>...<j0195> 2
<U4E00>..<U9FA5> 2
END WIDTH
WIDTH_DEFAULT 1
```

In this example, the code point values represented by <A> and <B> are assigned a width of 1. The code point values <j0101>...<j0195> (decimal ellipses) and <U4E00>..<U9FA5> are assigned a width of 2. The last line defines the DEFAULT_WIDTH to 1.

6 REPERTOIREMAP

FDCC-set and Charmap sources may be specified in a coded character set independent way, using symbolic character names. The relation between the symbolic character names and characters may be specified via a Repertoiremap, which defines the repertoire of characters defined for a FDCC-set, and the symbolic character names and corresponding abstract character (by a reference to ISO/IEC 10646).

The repertoire mapping is defined by specifying the symbolic character name and the ISO/IEC 10646 code position in hexadecimal form (with a preceding 'U') and optionally the long ISO/IEC 10646 character name in the following syntax:

"%s %s %s\n",<symbolic-name>,<short-identifier>,<comments>

The symbolic character name and the short identifier are each surrounded by angle brackets <>, and the fields are separated by one or more spaces or tabs on a line. If a right angle bracket or an escape character is used within a symbolic name, it is preceded by the escape character. The short identifier is either a ISO/IEC 10646 short identifier, or, if that does not exists, a short identifier in the range <P0000>..<PFFFF> or <P00000000>..<P7FFFFFF>.

The escape character can be redefined from the default reverse solidus (\) with the first line of the Repertoiremap containing the string "escape_char" followed by one or more spaces or tabs and then the escape character.

Several symbolic character names can refer to the same abstract character,

and are then used as synonyms in FDCC-sets and charmaps. The set of <U0000>..<UFFFF> and <U00000000>..<U7FFFFFFF> symbolic names (no lowercase letters) are predefined and refer to the corresponding code points of ISO/IEC 10646 with the same short identifier.

The "i18nrep" repertoiremap is defined to accommodate prior art, such as defined in Annex G of the ISO/IEC 9945-2:1993 standard, and used by ISO and IEC member bodies in their national POSIX locale specifications, and as used in POSIX locales distributed by the ISO/IEC POSIX working group and The Open Group. Many POSIX charmaps registered with ISO/IEC 15897 use these symbolic names. It also reflects use on the Internet, and many of the Internet registered charsets are specified using these symbolic names. The "i18nrep" repertoiremap thus facilitates reuse of both POSIX locale data and POSIX charmaps with data from this International Standard. The sequence <a8>..<z8> are used as hooks for tailoring to denote the last accented Latin letter of each of the ISO/IEC 646 letters <a>..<z>, so that tailorings that need to have specifications after the last letter of such a family, for example to introduce a new letter of an alphabet, can do so with a reference that is stable over different versions of the "i18n" FDCC-set. The "i18nrep" repertoiremap is not intended to cover the complete repertoire that the "i18n" FDCC-set covers. The contents of the "i18nrep" repertoiremap is as follows:

```
escape_char /
<NUL>                    <U0000>  NULL (NUL)
<SOH>                    <U0001>  START OF HEADING (SOH)
<STX>                    <U0002>  START OF TEXT (STX)
<ETX>                    <U0003>  END OF TEXT (ETX)
<EOT>                    <U0004>  END OF TRANSMISSION (EOT)
<ENQ>                    <U0005>  ENQUIRY (ENQ)
<ACK>                    <U0006>  ACKNOWLEDGE (ACK)
<alert>                  <U0007>  BELL (BEL)
<BEL>                    <U0007>  BELL (BEL)
<backspace>              <U0008>  BACKSPACE (BS)
<tab>                    <U0009>  CHARACTER TABULATION (HT)
<newline>                <U000A>  LINE FEED (LF)
<vertical-tab>           <U000B>  LINE TABULATION (VT)
<form-feed>              <U000C>  FORM FEED (FF)
<carriage-return>        <U000D>  CARRIAGE RETURN (CR)
<DLE>                    <U0010>  DATALINK ESCAPE (DLE)
<DC1>                    <U0011>  DEVICE CONTROL ONE (DC1)
<DC2>                    <U0012>  DEVICE CONTROL TWO (DC2)
<DC3>                    <U0013>  DEVICE CONTROL THREE (DC3)
<DC4>                    <U0014>  DEVICE CONTROL FOUR (DC4)
<NAK>                    <U0015>  NEGATIVE ACKNOWLEDGE (NAK)
<SYN>                    <U0016>  SYNCRONOUS IDLE (SYN)
<ETB>                    <U0017>  END OF TRANSMISSION BLOCK (ETB)
<CAN>                    <U0018>  CANCEL (CAN)
<SUB>                    <U001A>  SUBSTITUTE (SUB)
<ESC>                    <U001B>  ESCAPE (ESC)
<IS4>                    <U001C>  FILE SEPARATOR (IS4)
<IS3>                    <U001D>  GROUP SEPARATOR (IS3)
<intro>                  <U001D>  GROUP SEPARATOR (IS3)
<IS2>                    <U001E>  RECORD SEPARATOR (IS2)
<IS1>                    <U001F>  UNIT SEPARATOR (IS1)
<DEL>                    <U007F>  DELETE (DEL)
<space>                  <U0020>  SPACE
<exclamation-mark>       <U0021>  EXCLAMATION MARK
<quotation-mark>         <U0022>  QUOTATION MARK
<number-sign>            <U0023>  NUMBER SIGN
<dollar-sign>            <U0024>  DOLLAR SIGN
<percent-sign>           <U0025>  PERCENT SIGN
<ampersand>              <U0026>  AMPERSAND
```



```
<apostrophe>              <U0027>  APOSTROPHE
<left-parenthesis>        <U0028>  LEFT PARENTHESIS
<right-parenthesis>       <U0029>  RIGHT PARENTHESIS
<asterisk>                <U002A>  ASTERISK
<plus-sign>               <U002B>  PLUS SIGN
<comma>                   <U002C>  COMMA
<hyphen>                  <U002D>  HYPHEN-MINUS
<hyphen-minus>            <U002D>  HYPHEN-MINUS
<period>                  <U002E>  FULL STOP
<full-stop>               <U002E>  FULL STOP
<slash>                   <U002F>  SOLIDUS
<solidus>                 <U002F>  SOLIDUS
<zero>                    <U0030>  DIGIT ZERO
<one>                     <U0031>  DIGIT ONE
<two>                     <U0032>  DIGIT TWO
<three>                   <U0033>  DIGIT THREE
<four>                    <U0034>  DIGIT FOUR
<five>                    <U0035>  DIGIT FIVE
<six>                     <U0036>  DIGIT SIX
<seven>                   <U0037>  DIGIT SEVEN
<eight>                   <U0038>  DIGIT EIGHT
<nine>                    <U0039>  DIGIT NINE
<colon>                   <U003A>  COLON
<semicolon>               <U003B>  SEMICOLON
<less-than-sign>          <U003C>  LESS-THAN SIGN
<equals-sign>             <U003D>  EQUALS SIGN
<greater-than-sign>       <U003E>  GREATER-THAN SIGN
<question-mark>           <U003F>  QUESTION MARK
<commercial-at>           <U0040>  COMMERCIAL AT
<left-square-bracket>     <U005B>  LEFT SQUARE BRACKET
<backslash>               <U005C>  REVERSE SOLIDUS
<reverse-solidus>         <U005C>  REVERSE SOLIDUS
<right-square-bracket>    <U005D>  RIGHT SQUARE BRACKET
<circumflex>              <U005E>  CIRCUMFLEX ACCENT
<circumflex-accent>       <U005E>  CIRCUMFLEX ACCENT
<underscore>              <U005F>  LOW LINE
<low-line>                <U005F>  LOW LINE
<grave-accent>            <U0060>  GRAVE ACCENT
<left-brace>              <U007B>  LEFT CURLY BRACKET
<left-curly-bracket>      <U007B>  LEFT CURLY BRACKET
<vertical-line>           <U007C>  VERTICAL LINE
<right-brace>             <U007D>  RIGHT CURLY BRACKET
<right-curly-bracket>     <U007D>  RIGHT CURLY BRACKET
<tilde>                   <U007E>  TILDE

<a8>                      <P0001>  Weight indicating the position of the last a
<b8>                      <P0002>  Weight indicating the position of the last b
<c8>                      <P0003>  Weight indicating the position of the last c
<d8>                      <P0004>  Weight indicating the position of the last d
<e8>                      <P0005>  Weight indicating the position of the last e
<f8>                      <P0006>  Weight indicating the position of the last f
<g8>                      <P0007>  Weight indicating the position of the last g
<h8>                      <P0008>  Weight indicating the position of the last h
<i8>                      <P0009>  Weight indicating the position of the last i
<j8>                      <P0010>  Weight indicating the position of the last j
<k8>                      <P0011>  Weight indicating the position of the last k
<l8>                      <P0012>  Weight indicating the position of the last l
<m8>                      <P0013>  Weight indicating the position of the last m
<n8>                      <P0014>  Weight indicating the position of the last n
<o8>                      <P0015>  Weight indicating the position of the last o
<p8>                      <P0016>  Weight indicating the position of the last p
<q8>                      <P0017>  Weight indicating the position of the last q
<r8>                      <P0018>  Weight indicating the position of the last r
<s8>                      <P0019>  Weight indicating the position of the last s
<t8>                      <P0020>  Weight indicating the position of the last t
<u8>                      <P0021>  Weight indicating the position of the last u
<v8>                      <P0022>  Weight indicating the position of the last v
<w8>                      <P0023>  Weight indicating the position of the last w
<x8>                      <P0024>  Weight indicating the position of the last x
<y8>                      <P0025>  Weight indicating the position of the last y
<z8>                      <P0026>  Weight indicating the position of the last z

<NU>       <U0000>   NULL (NUL)
<SH>       <U0001>   START OF HEADING (SOH)
```



```
<SX>       <U0002>   START OF TEXT (STX)
<EX>       <U0003>   END OF TEXT (ETX)
<ET>       <U0004>   END OF TRANSMISSION (EOT)
<EQ>       <U0005>   ENQUIRY (ENQ)
<AK>       <U0006>   ACKNOWLEDGE (ACK)
<BL>       <U0007>   BELL (BEL)
<BS>       <U0008>   BACKSPACE (BS)
<HT>       <U0009>   CHARACTER TABULATION (HT)
<LF>       <U000A>   LINE FEED (LF)
<VT>       <U000B>   LINE TABULATION (VT)
<FF>       <U000C>   FORM FEED (FF)
<CR>       <U000D>   CARRIAGE RETURN (CR)
<SO>       <U000E>   SHIFT OUT (SO)
<SI>       <U000F>   SHIFT IN (SI)
<DL>       <U0010>   DATALINK ESCAPE (DLE)
<D1>       <U0011>   DEVICE CONTROL ONE (DC1)
<D2>       <U0012>   DEVICE CONTROL TWO (DC2)
<D3>       <U0013>   DEVICE CONTROL THREE (DC3)
<D4>       <U0014>   DEVICE CONTROL FOUR (DC4)
<NK>       <U0015>   NEGATIVE ACKNOWLEDGE (NAK)
<SY>       <U0016>   SYNCHRONOUS IDLE (SYN)
<EB>       <U0017>   END OF TRANSMISSION BLOCK (ETB)
<CN>       <U0018>   CANCEL (CAN)
<EM>       <U0019>   END OF MEDIUM (EM)
<SB>       <U001A>   SUBSTITUTE (SUB)
<EC>       <U001B>   ESCAPE (ESC)
<FS>       <U001C>   FILE SEPARATOR (IS4)
<GS>       <U001D>   GROUP SEPARATOR (IS3)
<RS>       <U001E>   RECORD SEPARATOR (IS2)
<US>       <U001F>   UNIT SEPARATOR (IS1)
<DT>       <U007F>   DELETE (DEL)
<PA>       <U0080>   PADDING CHARACTER (PAD)
<HO>       <U0081>   HIGH OCTET PRESET (HOP)
<BH>       <U0082>   BREAK PERMITTED HERE (BPH)
<NH>       <U0083>   NO BREAK HERE (NBH)
<IN>       <U0084>   INDEX (IND)
<NL>       <U0085>   NEXT LINE (NEL)
<SA>       <U0086>   START OF SELECTED AREA (SSA)
<ES>       <U0087>   END OF SELECTED AREA (ESA)
<HS>       <U0088>   CHARACTER TABULATION SET (HTS)
<HJ>       <U0089>   CHARACTER TABULATION WITH JUSTIFICATION (HTJ)
<VS>       <U008A>   LINE TABULATION SET (VTS)
<PD>       <U008B>   PARTIAL LINE FORWARD (PLD)
<PU>       <U008C>   PARTIAL LINE BACKWARD (PLU)
<RI>       <U008D>   REVERSE LINE FEED (RI)
<S2>       <U008E>   SINGLE-SHIFT TWO (SS2)
<S3>       <U008F>   SINGLE-SHIFT THREE (SS3)
<DC>       <U0090>   DEVICE CONTROL STRING (DCS)
<P1>       <U0091>   PRIVATE USE ONE (PU1)
<P2>       <U0092>   PRIVATE USE TWO (PU2)
<TS>       <U0093>   SET TRANSMIT STATE (STS)
<CC>       <U0094>   CANCEL CHARACTER (CCH)
<MW>       <U0095>   MESSAGE WAITING (MW)
<SG>       <U0096>   START OF GUARDED AREA (SPA)
<EG>       <U0097>   END OF GUARDED AREA (EPA)
<SS>       <U0098>   START OF STRING (SOS)
<GC>       <U0099>   SINGLE GRAPHIC CHARACTER INTRODUCER (SGCI)
<SC>       <U009A>   SINGLE CHARACTER INTRODUCER (SCI)
<CI>       <U009B>   CONTROL SEQUENCE INTRODUCER (CSI)
<ST>       <U009C>   STRING TERMINATOR (ST)
<OC>       <U009D>   OPERATING SYSTEM COMMAND (OSC)
<PM>       <U009E>   PRIVACY MESSAGE (PM)
<AC>       <U009F>   APPLICATION PROGRAM COMMAND (APC)
<SP>       <U0020>   SPACE
<!>        <U0021>   EXCLAMATION MARK
<">        <U0022>   QUOTATION MARK
<Nb>       <U0023>   NUMBER SIGN
<DO>       <U0024>   DOLLAR SIGN
<%>        <U0025>   PERCENT SIGN
<&>        <U0026>   AMPERSAND
<'>        <U0027>   APOSTROPHE
<(>        <U0028>   LEFT PARENTHESIS
<)>        <U0029>   RIGHT PARENTHESIS
<*>        <U002A>   ASTERISK
```



```
<+>        <U002B>   PLUS SIGN
<,>        <U002C>   COMMA
<->        <U002D>   HYPHEN-MINUS
<.>        <U002E>   FULL STOP
<//>       <U002F>   SOLIDUS
<0>        <U0030>   DIGIT ZERO
<1>        <U0031>   DIGIT ONE
<2>        <U0032>   DIGIT TWO
<3>        <U0033>   DIGIT THREE
<4>        <U0034>   DIGIT FOUR
<5>        <U0035>   DIGIT FIVE
<6>        <U0036>   DIGIT SIX
<7>        <U0037>   DIGIT SEVEN
<8>        <U0038>   DIGIT EIGHT
<9>        <U0039>   DIGIT NINE
<:>        <U003A>   COLON
<;>        <U003B>   SEMICOLON
<<>        <U003C>   LESS-THAN SIGN
<=>        <U003D>   EQUALS SIGN
</>>       <U003E>   GREATER-THAN SIGN
<?>        <U003F>   QUESTION MARK
<At>       <U0040>   COMMERCIAL AT
<A>        <U0041>   LATIN CAPITAL LETTER A
<B>        <U0042>   LATIN CAPITAL LETTER B
<C>        <U0043>   LATIN CAPITAL LETTER C
<D>        <U0044>   LATIN CAPITAL LETTER D
<E>        <U0045>   LATIN CAPITAL LETTER E
<F>        <U0046>   LATIN CAPITAL LETTER F
<G>        <U0047>   LATIN CAPITAL LETTER G
<H>        <U0048>   LATIN CAPITAL LETTER H
<I>        <U0049>   LATIN CAPITAL LETTER I
<J>        <U004A>   LATIN CAPITAL LETTER J
<K>        <U004B>   LATIN CAPITAL LETTER K
<L>        <U004C>   LATIN CAPITAL LETTER L
<M>        <U004D>   LATIN CAPITAL LETTER M
<N>        <U004E>   LATIN CAPITAL LETTER N
<O>        <U004F>   LATIN CAPITAL LETTER O
<P>        <U0050>   LATIN CAPITAL LETTER P
<Q>        <U0051>   LATIN CAPITAL LETTER Q
<R>        <U0052>   LATIN CAPITAL LETTER R
<S>        <U0053>   LATIN CAPITAL LETTER S
<T>        <U0054>   LATIN CAPITAL LETTER T
<U>        <U0055>   LATIN CAPITAL LETTER U
<V>        <U0056>   LATIN CAPITAL LETTER V
<W>        <U0057>   LATIN CAPITAL LETTER W
<X>        <U0058>   LATIN CAPITAL LETTER X
<Y>        <U0059>   LATIN CAPITAL LETTER Y
<Z>        <U005A>   LATIN CAPITAL LETTER Z
<<(>       <U005B>   LEFT SQUARE BRACKET
<////>     <U005C>   REVERSE SOLIDUS
<)/>>      <U005D>   RIGHT SQUARE BRACKET
<'/>>      <U005E>   CIRCUMFLEX ACCENT
<_>        <U005F>   LOW LINE
<'!>       <U0060>   GRAVE ACCENT
<a>        <U0061>   LATIN SMALL LETTER A
<b>        <U0062>   LATIN SMALL LETTER B
<c>        <U0063>   LATIN SMALL LETTER C
<d>        <U0064>   LATIN SMALL LETTER D
<e>        <U0065>   LATIN SMALL LETTER E
<f>        <U0066>   LATIN SMALL LETTER F
<g>        <U0067>   LATIN SMALL LETTER G
<h>        <U0068>   LATIN SMALL LETTER H
<i>        <U0069>   LATIN SMALL LETTER I
<j>        <U006A>   LATIN SMALL LETTER J
<k>        <U006B>   LATIN SMALL LETTER K
<l>        <U006C>   LATIN SMALL LETTER L
<m>        <U006D>   LATIN SMALL LETTER M
<n>        <U006E>   LATIN SMALL LETTER N
<o>        <U006F>   LATIN SMALL LETTER O
<p>        <U0070>   LATIN SMALL LETTER P
<q>        <U0071>   LATIN SMALL LETTER Q
<r>        <U0072>   LATIN SMALL LETTER R
<s>        <U0073>   LATIN SMALL LETTER S
<t>        <U0074>   LATIN SMALL LETTER T
```



```
<u>        <U0075>   LATIN SMALL LETTER U
<v>        <U0076>   LATIN SMALL LETTER V
<w>        <U0077>   LATIN SMALL LETTER W
<x>        <U0078>   LATIN SMALL LETTER X
<y>        <U0079>   LATIN SMALL LETTER Y
<z>        <U007A>   LATIN SMALL LETTER Z
<(!>       <U007B>   LEFT CURLY BRACKET
<!!>       <U007C>   VERTICAL LINE
<!)>       <U007D>   RIGHT CURLY BRACKET
<'?>       <U007E>   TILDE
<NS>       <U00A0>   NO-BREAK SPACE
<!I>       <U00A1>   INVERTED EXCLAMATION MARK
<Ct>       <U00A2>   CENT SIGN
<Pd>       <U00A3>   POUND SIGN
<Cu>       <U00A4>   CURRENCY SIGN
<Ye>       <U00A5>   YEN SIGN
<BB>       <U00A6>   BROKEN BAR
<SE>       <U00A7>   SECTION SIGN
<':>       <U00A8>   DIAERESIS
<Co>       <U00A9>   COPYRIGHT SIGN
<-a>       <U00AA>   FEMININE ORDINAL INDICATOR
<<<>       <U00AB>   LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
<NO>       <U00AC>   NOT SIGN
<-->       <U00AD>   SOFT HYPHEN
<Rg>       <U00AE>   REGISTERED SIGN
<'m>       <U00AF>   MACRON
<DG>       <U00B0>   DEGREE SIGN
<+->       <U00B1>   PLUS-MINUS SIGN
<2S>       <U00B2>   SUPERSCRIPT TWO
<3S>       <U00B3>   SUPERSCRIPT THREE
<''>       <U00B4>   ACUTE ACCENT
<My>       <U00B5>   MICRO SIGN
<PI>       <U00B6>   PILCROW SIGN
<.M>       <U00B7>   MIDDLE DOT
<',>       <U00B8>   CEDILLA
<1S>       <U00B9>   SUPERSCRIPT ONE
<-o>       <U00BA>   MASCULINE ORDINAL INDICATOR
</>/>>     <U00BB>   RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
<14>       <U00BC>   VULGAR FRACTION ONE QUARTER
<12>       <U00BD>   VULGAR FRACTION ONE HALF
<34>       <U00BE>   VULGAR FRACTION THREE QUARTERS
<?I>       <U00BF>   INVERTED QUESTION MARK
<A!>       <U00C0>   LATIN CAPITAL LETTER A WITH GRAVE
<A'>       <U00C1>   LATIN CAPITAL LETTER A WITH ACUTE
<A/>>      <U00C2>   LATIN CAPITAL LETTER A WITH CIRCUMFLEX
<A?>       <U00C3>   LATIN CAPITAL LETTER A WITH TILDE
<A:>       <U00C4>   LATIN CAPITAL LETTER A WITH DIAERESIS
<AA>       <U00C5>   LATIN CAPITAL LETTER A WITH RING ABOVE
<AE>       <U00C6>   LATIN CAPITAL LETTER AE (ash)
<C,>       <U00C7>   LATIN CAPITAL LETTER C WITH CEDILLA
<E!>       <U00C8>   LATIN CAPITAL LETTER E WITH GRAVE
<E'>       <U00C9>   LATIN CAPITAL LETTER E WITH ACUTE
<E/>>      <U00CA>   LATIN CAPITAL LETTER E WITH CIRCUMFLEX
<E:>       <U00CB>   LATIN CAPITAL LETTER E WITH DIAERESIS
<I!>       <U00CC>   LATIN CAPITAL LETTER I WITH GRAVE
<I'>       <U00CD>   LATIN CAPITAL LETTER I WITH ACUTE
<I/>>      <U00CE>   LATIN CAPITAL LETTER I WITH CIRCUMFLEX
<I:>       <U00CF>   LATIN CAPITAL LETTER I WITH DIAERESIS
<D->       <U00D0>   LATIN CAPITAL LETTER ETH (Icelandic)
<N?>       <U00D1>   LATIN CAPITAL LETTER N WITH TILDE
<O!>       <U00D2>   LATIN CAPITAL LETTER O WITH GRAVE
<O'>       <U00D3>   LATIN CAPITAL LETTER O WITH ACUTE
<O/>>      <U00D4>   LATIN CAPITAL LETTER O WITH CIRCUMFLEX
<O?>       <U00D5>   LATIN CAPITAL LETTER O WITH TILDE
<O:>       <U00D6>   LATIN CAPITAL LETTER O WITH DIAERESIS
<*X>       <U00D7>   MULTIPLICATION SIGN
<O//>      <U00D8>   LATIN CAPITAL LETTER O WITH STROKE
<U!>       <U00D9>   LATIN CAPITAL LETTER U WITH GRAVE
<U'>       <U00DA>   LATIN CAPITAL LETTER U WITH ACUTE
<U/>>      <U00DB>   LATIN CAPITAL LETTER U WITH CIRCUMFLEX
<U:>       <U00DC>   LATIN CAPITAL LETTER U WITH DIAERESIS
<Y'>       <U00DD>   LATIN CAPITAL LETTER Y WITH ACUTE
<TH>       <U00DE>   LATIN CAPITAL LETTER THORN (Icelandic)
<ss>       <U00DF>   LATIN SMALL LETTER SHARP S (German)
```



```
<a!>       <U00E0>   LATIN SMALL LETTER A WITH GRAVE
<a'>       <U00E1>   LATIN SMALL LETTER A WITH ACUTE
<a/>>      <U00E2>   LATIN SMALL LETTER A WITH CIRCUMFLEX
<a?>       <U00E3>   LATIN SMALL LETTER A WITH TILDE
<a:>       <U00E4>   LATIN SMALL LETTER A WITH DIAERESIS
<aa>       <U00E5>   LATIN SMALL LETTER A WITH RING ABOVE
<ae>       <U00E6>   LATIN SMALL LETTER AE (ash)
<c,>       <U00E7>   LATIN SMALL LETTER C WITH CEDILLA
<e!>       <U00E8>   LATIN SMALL LETTER E WITH GRAVE
<e'>       <U00E9>   LATIN SMALL LETTER E WITH ACUTE
<e/>>      <U00EA>   LATIN SMALL LETTER E WITH CIRCUMFLEX
<e:>       <U00EB>   LATIN SMALL LETTER E WITH DIAERESIS
<i!>       <U00EC>   LATIN SMALL LETTER I WITH GRAVE
<i'>       <U00ED>   LATIN SMALL LETTER I WITH ACUTE
<i/>>      <U00EE>   LATIN SMALL LETTER I WITH CIRCUMFLEX
<i:>       <U00EF>   LATIN SMALL LETTER I WITH DIAERESIS
<d->       <U00F0>   LATIN SMALL LETTER ETH (Icelandic)
<n?>       <U00F1>   LATIN SMALL LETTER N WITH TILDE
<o!>       <U00F2>   LATIN SMALL LETTER O WITH GRAVE
<o'>       <U00F3>   LATIN SMALL LETTER O WITH ACUTE
<o/>>      <U00F4>   LATIN SMALL LETTER O WITH CIRCUMFLEX
<o?>       <U00F5>   LATIN SMALL LETTER O WITH TILDE
<o:>       <U00F6>   LATIN SMALL LETTER O WITH DIAERESIS
<-:>       <U00F7>   DIVISION SIGN
<o//>      <U00F8>   LATIN SMALL LETTER O WITH STROKE
<u!>       <U00F9>   LATIN SMALL LETTER U WITH GRAVE
<u'>       <U00FA>   LATIN SMALL LETTER U WITH ACUTE
<u/>>      <U00FB>   LATIN SMALL LETTER U WITH CIRCUMFLEX
<u:>       <U00FC>   LATIN SMALL LETTER U WITH DIAERESIS
<y'>       <U00FD>   LATIN SMALL LETTER Y WITH ACUTE
<th>       <U00FE>   LATIN SMALL LETTER THORN (Icelandic)
<y:>       <U00FF>   LATIN SMALL LETTER Y WITH DIAERESIS
<A->       <U0100>   LATIN CAPITAL LETTER A WITH MACRON
<a->       <U0101>   LATIN SMALL LETTER A WITH MACRON
<A(>       <U0102>   LATIN CAPITAL LETTER A WITH BREVE
<a(>       <U0103>   LATIN SMALL LETTER A WITH BREVE
<A;>       <U0104>   LATIN CAPITAL LETTER A WITH OGONEK
<a;>       <U0105>   LATIN SMALL LETTER A WITH OGONEK
<C'>       <U0106>   LATIN CAPITAL LETTER C WITH ACUTE
<c'>       <U0107>   LATIN SMALL LETTER C WITH ACUTE
<C/>>      <U0108>   LATIN CAPITAL LETTER C WITH CIRCUMFLEX
<c/>>      <U0109>   LATIN SMALL LETTER C WITH CIRCUMFLEX
<C.>       <U010A>   LATIN CAPITAL LETTER C WITH DOT ABOVE
<c.>       <U010B>   LATIN SMALL LETTER C WITH DOT ABOVE
<C<>       <U010C>   LATIN CAPITAL LETTER C WITH CARON
<c<>       <U010D>   LATIN SMALL LETTER C WITH CARON
<D<>       <U010E>   LATIN CAPITAL LETTER D WITH CARON
<d<>       <U010F>   LATIN SMALL LETTER D WITH CARON
<D//>      <U0110>   LATIN CAPITAL LETTER D WITH STROKE
<d//>      <U0111>   LATIN SMALL LETTER D WITH STROKE
<E->       <U0112>   LATIN CAPITAL LETTER E WITH MACRON
<e->       <U0113>   LATIN SMALL LETTER E WITH MACRON
<E(>       <U0114>   LATIN CAPITAL LETTER E WITH BREVE
<e(>       <U0115>   LATIN SMALL LETTER E WITH BREVE
<E.>       <U0116>   LATIN CAPITAL LETTER E WITH DOT ABOVE
<e.>       <U0117>   LATIN SMALL LETTER E WITH DOT ABOVE
<E;>       <U0118>   LATIN CAPITAL LETTER E WITH OGONEK
<e;>       <U0119>   LATIN SMALL LETTER E WITH OGONEK
<E<>       <U011A>   LATIN CAPITAL LETTER E WITH CARON
<e<>       <U011B>   LATIN SMALL LETTER E WITH CARON
<G/>>      <U011C>   LATIN CAPITAL LETTER G WITH CIRCUMFLEX
<g/>>      <U011D>   LATIN SMALL LETTER G WITH CIRCUMFLEX
<G(>       <U011E>   LATIN CAPITAL LETTER G WITH BREVE
<g(>       <U011F>   LATIN SMALL LETTER G WITH BREVE
<G.>       <U0120>   LATIN CAPITAL LETTER G WITH DOT ABOVE
<g.>       <U0121>   LATIN SMALL LETTER G WITH DOT ABOVE
<G,>       <U0122>   LATIN CAPITAL LETTER G WITH CEDILLA
<g,>       <U0123>   LATIN SMALL LETTER G WITH CEDILLA
<H/>>      <U0124>   LATIN CAPITAL LETTER H WITH CIRCUMFLEX
<h/>>      <U0125>   LATIN SMALL LETTER H WITH CIRCUMFLEX
<H//>      <U0126>   LATIN CAPITAL LETTER H WITH STROKE
<h//>      <U0127>   LATIN SMALL LETTER H WITH STROKE
<I?>       <U0128>   LATIN CAPITAL LETTER I WITH TILDE
<i?>       <U0129>   LATIN SMALL LETTER I WITH TILDE
```



```
<I->       <U012A>   LATIN CAPITAL LETTER I WITH MACRON
<i->       <U012B>   LATIN SMALL LETTER I WITH MACRON
<I(>       <U012C>   LATIN CAPITAL LETTER I WITH BREVE
<i(>       <U012D>   LATIN SMALL LETTER I WITH BREVE
<I;>       <U012E>   LATIN CAPITAL LETTER I WITH OGONEK
<i;>       <U012F>   LATIN SMALL LETTER I WITH OGONEK
<I.>       <U0130>   LATIN CAPITAL LETTER I WITH DOT ABOVE
<i.>       <U0131>   LATIN SMALL LETTER DOTLESS I
<IJ>       <U0132>   LATIN CAPITAL LIGATURE IJ
<ij>       <U0133>   LATIN SMALL LIGATURE IJ
<J/>>      <U0134>   LATIN CAPITAL LETTER J WITH CIRCUMFLEX
<j/>>      <U0135>   LATIN SMALL LETTER J WITH CIRCUMFLEX
<K,>       <U0136>   LATIN CAPITAL LETTER K WITH CEDILLA
<k,>       <U0137>   LATIN SMALL LETTER K WITH CEDILLA
<kk>       <U0138>   LATIN SMALL LETTER KRA (Greenlandic)
<L'>       <U0139>   LATIN CAPITAL LETTER L WITH ACUTE
<l'>       <U013A>   LATIN SMALL LETTER L WITH ACUTE
<L,>       <U013B>   LATIN CAPITAL LETTER L WITH CEDILLA
<l,>       <U013C>   LATIN SMALL LETTER L WITH CEDILLA
<L<>       <U013D>   LATIN CAPITAL LETTER L WITH CARON
<l<>       <U013E>   LATIN SMALL LETTER L WITH CARON
<L.>       <U013F>   LATIN CAPITAL LETTER L WITH MIDDLE DOT
<l.>       <U0140>   LATIN SMALL LETTER L WITH MIDDLE DOT
<L//>      <U0141>   LATIN CAPITAL LETTER L WITH STROKE
<l//>      <U0142>   LATIN SMALL LETTER L WITH STROKE
<N'>       <U0143>   LATIN CAPITAL LETTER N WITH ACUTE
<n'>       <U0144>   LATIN SMALL LETTER N WITH ACUTE
<N,>       <U0145>   LATIN CAPITAL LETTER N WITH CEDILLA
<n,>       <U0146>   LATIN SMALL LETTER N WITH CEDILLA
<N<>       <U0147>   LATIN CAPITAL LETTER N WITH CARON
<n<>       <U0148>   LATIN SMALL LETTER N WITH CARON
<'n>       <U0149>   LATIN SMALL LETTER N PRECEDED BY APOSTROPHE
<NG>       <U014A>   LATIN CAPITAL LETTER ENG (Sami)
<ng>       <U014B>   LATIN SMALL LETTER ENG (Sami)
<O->       <U014C>   LATIN CAPITAL LETTER O WITH MACRON
<o->       <U014D>   LATIN SMALL LETTER O WITH MACRON
<O(>       <U014E>   LATIN CAPITAL LETTER O WITH BREVE
<o(>       <U014F>   LATIN SMALL LETTER O WITH BREVE
<O">       <U0150>   LATIN CAPITAL LETTER O WITH DOUBLE ACUTE
<o">       <U0151>   LATIN SMALL LETTER O WITH DOUBLE ACUTE
<OE>       <U0152>   LATIN CAPITAL LIGATURE OE
<oe>       <U0153>   LATIN SMALL LIGATURE OE
<R'>       <U0154>   LATIN CAPITAL LETTER R WITH ACUTE
<r'>       <U0155>   LATIN SMALL LETTER R WITH ACUTE
<R,>       <U0156>   LATIN CAPITAL LETTER R WITH CEDILLA
<r,>       <U0157>   LATIN SMALL LETTER R WITH CEDILLA
<R<>       <U0158>   LATIN CAPITAL LETTER R WITH CARON
<r<>       <U0159>   LATIN SMALL LETTER R WITH CARON
<S'>       <U015A>   LATIN CAPITAL LETTER S WITH ACUTE
<s'>       <U015B>   LATIN SMALL LETTER S WITH ACUTE
<S/>>      <U015C>   LATIN CAPITAL LETTER S WITH CIRCUMFLEX
<s/>>      <U015D>   LATIN SMALL LETTER S WITH CIRCUMFLEX
<S,>       <U015E>   LATIN CAPITAL LETTER S WITH CEDILLA
<s,>       <U015F>   LATIN SMALL LETTER S WITH CEDILLA
<S<>       <U0160>   LATIN CAPITAL LETTER S WITH CARON
<s<>       <U0161>   LATIN SMALL LETTER S WITH CARON
<T,>       <U0162>   LATIN CAPITAL LETTER T WITH CEDILLA
<t,>       <U0163>   LATIN SMALL LETTER T WITH CEDILLA
<T<>       <U0164>   LATIN CAPITAL LETTER T WITH CARON
<t<>       <U0165>   LATIN SMALL LETTER T WITH CARON
<T//>      <U0166>   LATIN CAPITAL LETTER T WITH STROKE
<t//>      <U0167>   LATIN SMALL LETTER T WITH STROKE
<U?>       <U0168>   LATIN CAPITAL LETTER U WITH TILDE
<u?>       <U0169>   LATIN SMALL LETTER U WITH TILDE
<U->       <U016A>   LATIN CAPITAL LETTER U WITH MACRON
<u->       <U016B>   LATIN SMALL LETTER U WITH MACRON
<U(>       <U016C>   LATIN CAPITAL LETTER U WITH BREVE
<u(>       <U016D>   LATIN SMALL LETTER U WITH BREVE
<U0>       <U016E>   LATIN CAPITAL LETTER U WITH RING ABOVE
<u0>       <U016F>   LATIN SMALL LETTER U WITH RING ABOVE
<U">       <U0170>   LATIN CAPITAL LETTER U WITH DOUBLE ACUTE
<u">       <U0171>   LATIN SMALL LETTER U WITH DOUBLE ACUTE
<U;>       <U0172>   LATIN CAPITAL LETTER U WITH OGONEK
<u;>       <U0173>   LATIN SMALL LETTER U WITH OGONEK
```



```
<W/>>      <U0174>   LATIN CAPITAL LETTER W WITH CIRCUMFLEX
<w/>>      <U0175>   LATIN SMALL LETTER W WITH CIRCUMFLEX
<Y/>>      <U0176>   LATIN CAPITAL LETTER Y WITH CIRCUMFLEX
<y/>>      <U0177>   LATIN SMALL LETTER Y WITH CIRCUMFLEX
<Y:>       <U0178>   LATIN CAPITAL LETTER Y WITH DIAERESIS
<Z'>       <U0179>   LATIN CAPITAL LETTER Z WITH ACUTE
<z'>       <U017A>   LATIN SMALL LETTER Z WITH ACUTE
<Z.>       <U017B>   LATIN CAPITAL LETTER Z WITH DOT ABOVE
<z.>       <U017C>   LATIN SMALL LETTER Z WITH DOT ABOVE
<Z<>       <U017D>   LATIN CAPITAL LETTER Z WITH CARON
<z<>       <U017E>   LATIN SMALL LETTER Z WITH CARON
<s1>       <U017F>   LATIN SMALL LETTER LONG S
<b//>      <U0180>   LATIN SMALL LETTER B WITH STROKE
<B2>       <U0181>   LATIN CAPITAL LETTER B WITH HOOK
<C2>       <U0187>   LATIN CAPITAL LETTER C WITH HOOK
<c2>       <U0188>   LATIN SMALL LETTER C WITH HOOK
<F2>       <U0191>   LATIN CAPITAL LETTER F WITH HOOK
<f2>       <U0192>   LATIN SMALL LETTER F WITH HOOK
<K2>       <U0198>   LATIN CAPITAL LETTER K WITH HOOK
<k2>       <U0199>   LATIN SMALL LETTER K WITH HOOK
<O9>       <U01A0>   LATIN CAPITAL LETTER O WITH HORN
<o9>       <U01A1>   LATIN SMALL LETTER O WITH HORN
<OI>       <U01A2>   LATIN CAPITAL LETTER OI
<oi>       <U01A3>   LATIN SMALL LETTER OI
<yr>       <U01A6>   LATIN LETTER YR
<U9>       <U01AF>   LATIN CAPITAL LETTER U WITH HORN
<u9>       <U01B0>   LATIN SMALL LETTER U WITH HORN
<Z//>      <U01B5>   LATIN CAPITAL LETTER Z WITH STROKE
<z//>      <U01B6>   LATIN SMALL LETTER Z WITH STROKE
<ED>       <U01B7>   LATIN CAPITAL LETTER EZH
<DZ<>      <U01C4>   LATIN CAPITAL LETTER DZ WITH CARON
<Dz<>      <U01C5>   LATIN CAPITAL LETTER D WITH SMALL LETTER Z WITH CARON
<dz<>      <U01C6>   LATIN SMALL LETTER DZ WITH CARON
<LJ3>      <U01C7>   LATIN CAPITAL LETTER LJ
<Lj3>      <U01C8>   LATIN CAPITAL LETTER L WITH SMALL LETTER J
<lj3>      <U01C9>   LATIN SMALL LETTER LJ
<NJ3>      <U01CA>   LATIN CAPITAL LETTER NJ
<Nj3>      <U01CB>   LATIN CAPITAL LETTER N WITH SMALL LETTER J
<nj3>      <U01CC>   LATIN SMALL LETTER NJ
<A<>       <U01CD>   LATIN CAPITAL LETTER A WITH CARON
<a<>       <U01CE>   LATIN SMALL LETTER A WITH CARON
<I<>       <U01CF>   LATIN CAPITAL LETTER I WITH CARON
<i<>       <U01D0>   LATIN SMALL LETTER I WITH CARON
<O<>       <U01D1>   LATIN CAPITAL LETTER O WITH CARON
<o<>       <U01D2>   LATIN SMALL LETTER O WITH CARON
<U<>       <U01D3>   LATIN CAPITAL LETTER U WITH CARON
<u<>       <U01D4>   LATIN SMALL LETTER U WITH CARON
<U:->      <U01D5>   LATIN CAPITAL LETTER U WITH DIAERESIS AND MACRON
<u:->      <U01D6>   LATIN SMALL LETTER U WITH DIAERESIS AND MACRON
<U:'>      <U01D7>   LATIN CAPITAL LETTER U WITH DIAERESIS AND ACUTE
<u:'>      <U01D8>   LATIN SMALL LETTER U WITH DIAERESIS AND ACUTE
<U:<>      <U01D9>   LATIN CAPITAL LETTER U WITH DIAERESIS AND CARON
<u:<>      <U01DA>   LATIN SMALL LETTER U WITH DIAERESIS AND CARON
<U:!>      <U01DB>   LATIN CAPITAL LETTER U WITH DIAERESIS AND GRAVE
<u:!>      <U01DC>   LATIN SMALL LETTER U WITH DIAERESIS AND GRAVE
<e1>       <U01DD>   LATIN SMALL LETTER TURNED E
<A1>       <U01DE>   LATIN CAPITAL LETTER A WITH DIAERESIS AND MACRON
<a1>       <U01DF>   LATIN SMALL LETTER A WITH DIAERESIS AND MACRON
<A7>       <U01E0>   LATIN CAPITAL LETTER A WITH DOT ABOVE AND MACRON
<a7>       <U01E1>   LATIN SMALL LETTER A WITH DOT ABOVE AND MACRON
<A3>       <U01E2>   LATIN CAPITAL LETTER AE WITH MACRON (ash)
<a3>       <U01E3>   LATIN SMALL LETTER AE WITH MACRON (ash)
<G//>      <U01E4>   LATIN CAPITAL LETTER G WITH STROKE
<g//>      <U01E5>   LATIN SMALL LETTER G WITH STROKE
<G<>       <U01E6>   LATIN CAPITAL LETTER G WITH CARON
<g<>       <U01E7>   LATIN SMALL LETTER G WITH CARON
<K<>       <U01E8>   LATIN CAPITAL LETTER K WITH CARON
<k<>       <U01E9>   LATIN SMALL LETTER K WITH CARON
<O;>       <U01EA>   LATIN CAPITAL LETTER O WITH OGONEK
<o;>       <U01EB>   LATIN SMALL LETTER O WITH OGONEK
<O1>       <U01EC>   LATIN CAPITAL LETTER O WITH OGONEK AND MACRON
<o1>       <U01ED>   LATIN SMALL LETTER O WITH OGONEK AND MACRON
<EZ>       <U01EE>   LATIN CAPITAL LETTER EZH WITH CARON
<ez>       <U01EF>   LATIN SMALL LETTER EZH WITH CARON
```



```
<j<>       <U01F0>   LATIN SMALL LETTER J WITH CARON
<DZ3>      <U01F1>   LATIN CAPITAL LETTER DZ
<Dz3>      <U01F2>   LATIN CAPITAL LETTER D WITH SMALL LETTER Z
<dz3>      <U01F3>   LATIN SMALL LETTER DZ
<G'>       <U01F4>   LATIN CAPITAL LETTER G WITH ACUTE
<g'>       <U01F5>   LATIN SMALL LETTER G WITH ACUTE
<AA'>      <U01FA>   LATIN CAPITAL LETTER A WITH RING ABOVE AND ACUTE
<aa'>      <U01FB>   LATIN SMALL LETTER A WITH RING ABOVE AND ACUTE
<AE'>      <U01FC>   LATIN CAPITAL LETTER AE WITH ACUTE (ash)
<ae'>      <U01FD>   LATIN SMALL LETTER AE WITH ACUTE (ash)
<O//'>     <U01FE>   LATIN CAPITAL LETTER O WITH STROKE AND ACUTE
<o//'>     <U01FF>   LATIN SMALL LETTER O WITH STROKE AND ACUTE
<A!!>      <U0200>   LATIN CAPITAL LETTER A WITH DOUBLE GRAVE
<a!!>      <U0201>   LATIN SMALL LETTER A WITH DOUBLE GRAVE
<A)>       <U0202>   LATIN CAPITAL LETTER A WITH INVERTED BREVE
<a)>       <U0203>   LATIN SMALL LETTER A WITH INVERTED BREVE
<E!!>      <U0204>   LATIN CAPITAL LETTER E WITH DOUBLE GRAVE
<e!!>      <U0205>   LATIN SMALL LETTER E WITH DOUBLE GRAVE
<E)>       <U0206>   LATIN CAPITAL LETTER E WITH INVERTED BREVE
<e)>       <U0207>   LATIN SMALL LETTER E WITH INVERTED BREVE
<I!!>      <U0208>   LATIN CAPITAL LETTER I WITH DOUBLE GRAVE
<i!!>      <U0209>   LATIN SMALL LETTER I WITH DOUBLE GRAVE
<I)>       <U020A>   LATIN CAPITAL LETTER I WITH INVERTED BREVE
<i)>       <U020B>   LATIN SMALL LETTER I WITH INVERTED BREVE
<O!!>      <U020C>   LATIN CAPITAL LETTER O WITH DOUBLE GRAVE
<o!!>      <U020D>   LATIN SMALL LETTER O WITH DOUBLE GRAVE
<O)>       <U020E>   LATIN CAPITAL LETTER O WITH INVERTED BREVE
<o)>       <U020F>   LATIN SMALL LETTER O WITH INVERTED BREVE
<R!!>      <U0210>   LATIN CAPITAL LETTER R WITH DOUBLE GRAVE
<r!!>      <U0211>   LATIN SMALL LETTER R WITH DOUBLE GRAVE
<R)>       <U0212>   LATIN CAPITAL LETTER R WITH INVERTED BREVE
<r)>       <U0213>   LATIN SMALL LETTER R WITH INVERTED BREVE
<U!!>      <U0214>   LATIN CAPITAL LETTER U WITH DOUBLE GRAVE
<u!!>      <U0215>   LATIN SMALL LETTER U WITH DOUBLE GRAVE
<U)>       <U0216>   LATIN CAPITAL LETTER U WITH INVERTED BREVE
<u)>       <U0217>   LATIN SMALL LETTER U WITH INVERTED BREVE
<r1>       <U027C>   LATIN SMALL LETTER R WITH LONG LEG
<ed>       <U0292>   LATIN SMALL LETTER EZH
<;S>       <U02BB>   MODIFIER LETTER TURNED COMMA
<1/>>      <U02C6>   MODIFIER LETTER CIRCUMFLEX ACCENT
<'<>       <U02C7>   CARON (Mandarin Chinese third tone)
<1->       <U02C9>   MODIFIER LETTER MACRON (Mandarin Chinese first tone)
<1!>       <U02CB>   MODIFIER LETTER GRAVE ACCENT (Mandarin Chinese fourth tone)
<'(>       <U02D8>   BREVE
<'.>       <U02D9>   DOT ABOVE (Mandarin Chinese light tone)
<'0>       <U02DA>   RING ABOVE
<';>       <U02DB>   OGONEK
<1?>       <U02DC>   SMALL TILDE
<'">       <U02DD>   DOUBLE ACUTE ACCENT
<'G>       <U0374>   GREEK NUMERAL SIGN (Dexia keraia)
<,G>       <U0375>   GREEK LOWER NUMERAL SIGN (Aristeri keraia)
<j3>       <U037A>   GREEK YPOGEGRAMMENI
<?%>       <U037E>   GREEK QUESTION MARK (Erotimatiko)
<'*>       <U0384>   GREEK TONOS
<'%>       <U0385>   GREEK DIALYTIKA TONOS
<A%>       <U0386>   GREEK CAPITAL LETTER ALPHA WITH TONOS
<.*>       <U0387>   GREEK ANO TELEIA
<E%>       <U0388>   GREEK CAPITAL LETTER EPSILON WITH TONOS
<Y%>       <U0389>   GREEK CAPITAL LETTER ETA WITH TONOS
<I%>       <U038A>   GREEK CAPITAL LETTER IOTA WITH TONOS
<O%>       <U038C>   GREEK CAPITAL LETTER OMICRON WITH TONOS
<U%>       <U038E>   GREEK CAPITAL LETTER UPSILON WITH TONOS
<W%>       <U038F>   GREEK CAPITAL LETTER OMEGA WITH TONOS
<i3>       <U0390>   GREEK SMALL LETTER IOTA WITH DIALYTIKA AND TONOS
<A*>       <U0391>   GREEK CAPITAL LETTER ALPHA
<B*>       <U0392>   GREEK CAPITAL LETTER BETA
<G*>       <U0393>   GREEK CAPITAL LETTER GAMMA
<D*>       <U0394>   GREEK CAPITAL LETTER DELTA
<E*>       <U0395>   GREEK CAPITAL LETTER EPSILON
<Z*>       <U0396>   GREEK CAPITAL LETTER ZETA
<Y*>       <U0397>   GREEK CAPITAL LETTER ETA
<H*>       <U0398>   GREEK CAPITAL LETTER THETA
<I*>       <U0399>   GREEK CAPITAL LETTER IOTA
<K*>       <U039A>   GREEK CAPITAL LETTER KAPPA
```



```
<L*>       <U039B>   GREEK CAPITAL LETTER LAMDA
<M*>       <U039C>   GREEK CAPITAL LETTER MU
<N*>       <U039D>   GREEK CAPITAL LETTER NU
<C*>       <U039E>   GREEK CAPITAL LETTER XI
<O*>       <U039F>   GREEK CAPITAL LETTER OMICRON
<P*>       <U03A0>   GREEK CAPITAL LETTER PI
<R*>       <U03A1>   GREEK CAPITAL LETTER RHO
<S*>       <U03A3>   GREEK CAPITAL LETTER SIGMA
<T*>       <U03A4>   GREEK CAPITAL LETTER TAU
<U*>       <U03A5>   GREEK CAPITAL LETTER UPSILON
<F*>       <U03A6>   GREEK CAPITAL LETTER PHI
<X*>       <U03A7>   GREEK CAPITAL LETTER CHI
<Q*>       <U03A8>   GREEK CAPITAL LETTER PSI
<W*>       <U03A9>   GREEK CAPITAL LETTER OMEGA
<J*>       <U03AA>   GREEK CAPITAL LETTER IOTA WITH DIALYTIKA
<V*>       <U03AB>   GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA
<a%>       <U03AC>   GREEK SMALL LETTER ALPHA WITH TONOS
<e%>       <U03AD>   GREEK SMALL LETTER EPSILON WITH TONOS
<y%>       <U03AE>   GREEK SMALL LETTER ETA WITH TONOS
<i%>       <U03AF>   GREEK SMALL LETTER IOTA WITH TONOS
<u3>       <U03B0>   GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND TONOS
<a*>       <U03B1>   GREEK SMALL LETTER ALPHA
<b*>       <U03B2>   GREEK SMALL LETTER BETA
<g*>       <U03B3>   GREEK SMALL LETTER GAMMA
<d*>       <U03B4>   GREEK SMALL LETTER DELTA
<e*>       <U03B5>   GREEK SMALL LETTER EPSILON
<z*>       <U03B6>   GREEK SMALL LETTER ZETA
<y*>       <U03B7>   GREEK SMALL LETTER ETA
<h*>       <U03B8>   GREEK SMALL LETTER THETA
<i*>       <U03B9>   GREEK SMALL LETTER IOTA
<k*>       <U03BA>   GREEK SMALL LETTER KAPPA
<l*>       <U03BB>   GREEK SMALL LETTER LAMDA
<m*>       <U03BC>   GREEK SMALL LETTER MU
<n*>       <U03BD>   GREEK SMALL LETTER NU
<c*>       <U03BE>   GREEK SMALL LETTER XI
<o*>       <U03BF>   GREEK SMALL LETTER OMICRON
<p*>       <U03C0>   GREEK SMALL LETTER PI
<r*>       <U03C1>   GREEK SMALL LETTER RHO
<*s>       <U03C2>   GREEK SMALL LETTER FINAL SIGMA
<s*>       <U03C3>   GREEK SMALL LETTER SIGMA
<t*>       <U03C4>   GREEK SMALL LETTER TAU
<u*>       <U03C5>   GREEK SMALL LETTER UPSILON
<f*>       <U03C6>   GREEK SMALL LETTER PHI
<x*>       <U03C7>   GREEK SMALL LETTER CHI
<q*>       <U03C8>   GREEK SMALL LETTER PSI
<w*>       <U03C9>   GREEK SMALL LETTER OMEGA
<j*>       <U03CA>   GREEK SMALL LETTER IOTA WITH DIALYTIKA
<v*>       <U03CB>   GREEK SMALL LETTER UPSILON WITH DIALYTIKA
<o%>       <U03CC>   GREEK SMALL LETTER OMICRON WITH TONOS
<u%>       <U03CD>   GREEK SMALL LETTER UPSILON WITH TONOS
<w%>       <U03CE>   GREEK SMALL LETTER OMEGA WITH TONOS
<b3>       <U03D0>   GREEK BETA SYMBOL
<T3>       <U03DA>   GREEK LETTER STIGMA
<M3>       <U03DC>   GREEK LETTER DIGAMMA
<K3>       <U03DE>   GREEK LETTER KOPPA
<P3>       <U03E0>   GREEK LETTER SAMPI
<IO>       <U0401>   CYRILLIC CAPITAL LETTER IO
<D%>       <U0402>   CYRILLIC CAPITAL LETTER DJE (Serbocroatian)
<G%>       <U0403>   CYRILLIC CAPITAL LETTER GJE
<IE>       <U0404>   CYRILLIC CAPITAL LETTER UKRAINIAN IE
<DS>       <U0405>   CYRILLIC CAPITAL LETTER DZE
<II>       <U0406>   CYRILLIC CAPITAL LETTER BYELORUSSIAN-UKRAINIAN I
<YI>       <U0407>   CYRILLIC CAPITAL LETTER YI (Ukrainian)
<J%>       <U0408>   CYRILLIC CAPITAL LETTER JE
<LJ>       <U0409>   CYRILLIC CAPITAL LETTER LJE
<NJ>       <U040A>   CYRILLIC CAPITAL LETTER NJE
<Ts>       <U040B>   CYRILLIC CAPITAL LETTER TSHE (Serbocroatian)
<KJ>       <U040C>   CYRILLIC CAPITAL LETTER KJE
<V%>       <U040E>   CYRILLIC CAPITAL LETTER SHORT U (Byelorussian)
<DZ>       <U040F>   CYRILLIC CAPITAL LETTER DZHE
<A=>       <U0410>   CYRILLIC CAPITAL LETTER A
<B=>       <U0411>   CYRILLIC CAPITAL LETTER BE
<V=>       <U0412>   CYRILLIC CAPITAL LETTER VE
<G=>       <U0413>   CYRILLIC CAPITAL LETTER GHE
```



```
<D=>       <U0414>   CYRILLIC CAPITAL LETTER DE
<E=>       <U0415>   CYRILLIC CAPITAL LETTER IE
<Z%>       <U0416>   CYRILLIC CAPITAL LETTER ZHE
<Z=>       <U0417>   CYRILLIC CAPITAL LETTER ZE
<I=>       <U0418>   CYRILLIC CAPITAL LETTER I
<J=>       <U0419>   CYRILLIC CAPITAL LETTER SHORT I
<K=>       <U041A>   CYRILLIC CAPITAL LETTER KA
<L=>       <U041B>   CYRILLIC CAPITAL LETTER EL
<M=>       <U041C>   CYRILLIC CAPITAL LETTER EM
<N=>       <U041D>   CYRILLIC CAPITAL LETTER EN
<O=>       <U041E>   CYRILLIC CAPITAL LETTER O
<P=>       <U041F>   CYRILLIC CAPITAL LETTER PE
<R=>       <U0420>   CYRILLIC CAPITAL LETTER ER
<S=>       <U0421>   CYRILLIC CAPITAL LETTER ES
<T=>       <U0422>   CYRILLIC CAPITAL LETTER TE
<U=>       <U0423>   CYRILLIC CAPITAL LETTER U
<F=>       <U0424>   CYRILLIC CAPITAL LETTER EF
<H=>       <U0425>   CYRILLIC CAPITAL LETTER HA
<C=>       <U0426>   CYRILLIC CAPITAL LETTER TSE
<C%>       <U0427>   CYRILLIC CAPITAL LETTER CHE
<S%>       <U0428>   CYRILLIC CAPITAL LETTER SHA
<Sc>       <U0429>   CYRILLIC CAPITAL LETTER SHCHA
<=">       <U042A>   CYRILLIC CAPITAL LETTER HARD SIGN
<Y=>       <U042B>   CYRILLIC CAPITAL LETTER YERU
<%">       <U042C>   CYRILLIC CAPITAL LETTER SOFT SIGN
<JE>       <U042D>   CYRILLIC CAPITAL LETTER E
<JU>       <U042E>   CYRILLIC CAPITAL LETTER YU
<JA>       <U042F>   CYRILLIC CAPITAL LETTER YA
<a=>       <U0430>   CYRILLIC SMALL LETTER A
<b=>       <U0431>   CYRILLIC SMALL LETTER BE
<v=>       <U0432>   CYRILLIC SMALL LETTER VE
<g=>       <U0433>   CYRILLIC SMALL LETTER GHE
<d=>       <U0434>   CYRILLIC SMALL LETTER DE
<e=>       <U0435>   CYRILLIC SMALL LETTER IE
<z%>       <U0436>   CYRILLIC SMALL LETTER ZHE
<z=>       <U0437>   CYRILLIC SMALL LETTER ZE
<i=>       <U0438>   CYRILLIC SMALL LETTER I
<j=>       <U0439>   CYRILLIC SMALL LETTER SHORT I
<k=>       <U043A>   CYRILLIC SMALL LETTER KA
<l=>       <U043B>   CYRILLIC SMALL LETTER EL
<m=>       <U043C>   CYRILLIC SMALL LETTER EM
<n=>       <U043D>   CYRILLIC SMALL LETTER EN
<o=>       <U043E>   CYRILLIC SMALL LETTER O
<p=>       <U043F>   CYRILLIC SMALL LETTER PE
<r=>       <U0440>   CYRILLIC SMALL LETTER ER
<s=>       <U0441>   CYRILLIC SMALL LETTER ES
<t=>       <U0442>   CYRILLIC SMALL LETTER TE
<u=>       <U0443>   CYRILLIC SMALL LETTER U
<f=>       <U0444>   CYRILLIC SMALL LETTER EF
<h=>       <U0445>   CYRILLIC SMALL LETTER HA
<c=>       <U0446>   CYRILLIC SMALL LETTER TSE
<c%>       <U0447>   CYRILLIC SMALL LETTER CHE
<s%>       <U0448>   CYRILLIC SMALL LETTER SHA
<sc>       <U0449>   CYRILLIC SMALL LETTER SHCHA
<='>       <U044A>   CYRILLIC SMALL LETTER HARD SIGN
<y=>       <U044B>   CYRILLIC SMALL LETTER YERU
<%'>       <U044C>   CYRILLIC SMALL LETTER SOFT SIGN
<je>       <U044D>   CYRILLIC SMALL LETTER E
<ju>       <U044E>   CYRILLIC SMALL LETTER YU
<ja>       <U044F>   CYRILLIC SMALL LETTER YA
<io>       <U0451>   CYRILLIC SMALL LETTER IO
<d%>       <U0452>   CYRILLIC SMALL LETTER DJE (Serbocroatian)
<g%>       <U0453>   CYRILLIC SMALL LETTER GJE
<ie>       <U0454>   CYRILLIC SMALL LETTER UKRAINIAN IE
<ds>       <U0455>   CYRILLIC SMALL LETTER DZE
<ii>       <U0456>   CYRILLIC SMALL LETTER BYELORUSSIAN-UKRAINIAN I
<yi>       <U0457>   CYRILLIC SMALL LETTER YI (Ukrainian)
<j%>       <U0458>   CYRILLIC SMALL LETTER JE
<lj>       <U0459>   CYRILLIC SMALL LETTER LJE
<nj>       <U045A>   CYRILLIC SMALL LETTER NJE
<ts>       <U045B>   CYRILLIC SMALL LETTER TSHE (Serbocroatian)
<kj>       <U045C>   CYRILLIC SMALL LETTER KJE
<v%>       <U045E>   CYRILLIC SMALL LETTER SHORT U (Byelorussian)
<dz>       <U045F>   CYRILLIC SMALL LETTER DZHE
```



```
<Y3>       <U0462>   CYRILLIC CAPITAL LETTER YAT
<y3>       <U0463>   CYRILLIC SMALL LETTER YAT
<O3>       <U046A>   CYRILLIC CAPITAL LETTER BIG YUS
<o3>       <U046B>   CYRILLIC SMALL LETTER BIG YUS
<F3>       <U0472>   CYRILLIC CAPITAL LETTER FITA
<f3>       <U0473>   CYRILLIC SMALL LETTER FITA
<V3>       <U0474>   CYRILLIC CAPITAL LETTER IZHITSA
<v3>       <U0475>   CYRILLIC SMALL LETTER IZHITSA
<C3>       <U0480>   CYRILLIC CAPITAL LETTER KOPPA
<c3>       <U0481>   CYRILLIC SMALL LETTER KOPPA
<G3>       <U0490>   CYRILLIC CAPITAL LETTER GHE WITH UPTURN
<g3>       <U0491>   CYRILLIC SMALL LETTER GHE WITH UPTURN
<A+>       <U05D0>   HEBREW LETTER ALEF
<B+>       <U05D1>   HEBREW LETTER BET
<G+>       <U05D2>   HEBREW LETTER GIMEL
<D+>       <U05D3>   HEBREW LETTER DALET
<H+>       <U05D4>   HEBREW LETTER HE
<W+>       <U05D5>   HEBREW LETTER VAV
<Z+>       <U05D6>   HEBREW LETTER ZAYIN
<X+>       <U05D7>   HEBREW LETTER HET
<Tj>       <U05D8>   HEBREW LETTER TET
<J+>       <U05D9>   HEBREW LETTER YOD
<K%>       <U05DA>   HEBREW LETTER FINAL KAF
<K+>       <U05DB>   HEBREW LETTER KAF
<L+>       <U05DC>   HEBREW LETTER LAMED
<M%>       <U05DD>   HEBREW LETTER FINAL MEM
<M+>       <U05DE>   HEBREW LETTER MEM
<N%>       <U05DF>   HEBREW LETTER FINAL NUN
<N+>       <U05E0>   HEBREW LETTER NUN
<S+>       <U05E1>   HEBREW LETTER SAMEKH
<E+>       <U05E2>   HEBREW LETTER AYIN
<P%>       <U05E3>   HEBREW LETTER FINAL PE
<P+>       <U05E4>   HEBREW LETTER PE
<Zj>       <U05E5>   HEBREW LETTER FINAL TSADI
<ZJ>       <U05E6>   HEBREW LETTER TSADI
<Q+>       <U05E7>   HEBREW LETTER QOF
<R+>       <U05E8>   HEBREW LETTER RESH
<Sh>       <U05E9>   HEBREW LETTER SHIN
<T+>       <U05EA>   HEBREW LETTER TAV
<,+>       <U060C>   ARABIC COMMA
<;+>       <U061B>   ARABIC SEMICOLON
<?+>       <U061F>   ARABIC QUESTION MARK
<H'>       <U0621>   ARABIC LETTER HAMZA
<aM>       <U0622>   ARABIC LETTER ALEF WITH MADDA ABOVE
<aH>       <U0623>   ARABIC LETTER ALEF WITH HAMZA ABOVE
<wH>       <U0624>   ARABIC LETTER WAW WITH HAMZA ABOVE
<ah>       <U0625>   ARABIC LETTER ALEF WITH HAMZA BELOW
<yH>       <U0626>   ARABIC LETTER YEH WITH HAMZA ABOVE
<a+>       <U0627>   ARABIC LETTER ALEF
<b+>       <U0628>   ARABIC LETTER BEH
<tm>       <U0629>   ARABIC LETTER TEH MARBUTA
<t+>       <U062A>   ARABIC LETTER TEH
<tk>       <U062B>   ARABIC LETTER THEH
<g+>       <U062C>   ARABIC LETTER JEEM
<hk>       <U062D>   ARABIC LETTER HAH
<x+>       <U062E>   ARABIC LETTER KHAH
<d+>       <U062F>   ARABIC LETTER DAL
<dk>       <U0630>   ARABIC LETTER THAL
<r+>       <U0631>   ARABIC LETTER REH
<z+>       <U0632>   ARABIC LETTER ZAIN
<s+>       <U0633>   ARABIC LETTER SEEN
<sn>       <U0634>   ARABIC LETTER SHEEN
<c+>       <U0635>   ARABIC LETTER SAD
<dd>       <U0636>   ARABIC LETTER DAD
<tj>       <U0637>   ARABIC LETTER TAH
<zH>       <U0638>   ARABIC LETTER ZAH
<e+>       <U0639>   ARABIC LETTER AIN
<i+>       <U063A>   ARABIC LETTER GHAIN
<++>       <U0640>   ARABIC TATWEEL
<f+>       <U0641>   ARABIC LETTER FEH
<q+>       <U0642>   ARABIC LETTER QAF
<k+>       <U0643>   ARABIC LETTER KAF
<l+>       <U0644>   ARABIC LETTER LAM
<m+>       <U0645>   ARABIC LETTER MEEM
```



```
<n+>       <U0646>   ARABIC LETTER NOON
<h+>       <U0647>   ARABIC LETTER HEH
<w+>       <U0648>   ARABIC LETTER WAW
<j+>       <U0649>   ARABIC LETTER ALEF MAKSURA
<y+>       <U064A>   ARABIC LETTER YEH
<:+>       <U064B>   ARABIC FATHATAN
<"+>       <U064C>   ARABIC DAMMATAN
<=+>       <U064D>   ARABIC KASRATAN
<//+>      <U064E>   ARABIC FATHA
<'+>       <U064F>   ARABIC DAMMA
<1+>       <U0650>   ARABIC KASRA
<3+>       <U0651>   ARABIC SHADDA
<0+>       <U0652>   ARABIC SUKUN
<0a>       <U0660>   ARABIC-INDIC DIGIT ZERO
<1a>       <U0661>   ARABIC-INDIC DIGIT ONE
<2a>       <U0662>   ARABIC-INDIC DIGIT TWO
<3a>       <U0663>   ARABIC-INDIC DIGIT THREE
<4a>       <U0664>   ARABIC-INDIC DIGIT FOUR
<5a>       <U0665>   ARABIC-INDIC DIGIT FIVE
<6a>       <U0666>   ARABIC-INDIC DIGIT SIX
<7a>       <U0667>   ARABIC-INDIC DIGIT SEVEN
<8a>       <U0668>   ARABIC-INDIC DIGIT EIGHT
<9a>       <U0669>   ARABIC-INDIC DIGIT NINE
<aS>       <U0670>   ARABIC LETTER SUPERSCRIPT ALEF
<p+>       <U067E>   ARABIC LETTER PEH
<hH>       <U0681>   ARABIC LETTER HAH WITH HAMZA ABOVE
<tc>       <U0686>   ARABIC LETTER TCHEH
<zj>       <U0698>   ARABIC LETTER JEH
<v+>       <U06A4>   ARABIC LETTER VEH
<gf>       <U06AF>   ARABIC LETTER GAF
<A-0>      <U1E00>   LATIN CAPITAL LETTER A WITH RING BELOW
<a-0>      <U1E01>   LATIN SMALL LETTER A WITH RING BELOW
<B.>       <U1E02>   LATIN CAPITAL LETTER B WITH DOT ABOVE
<b.>       <U1E03>   LATIN SMALL LETTER B WITH DOT ABOVE
<B-.>      <U1E04>   LATIN CAPITAL LETTER B WITH DOT BELOW
<b-.>      <U1E05>   LATIN SMALL LETTER B WITH DOT BELOW
<B_>       <U1E06>   LATIN CAPITAL LETTER B WITH LINE BELOW
<b_>       <U1E07>   LATIN SMALL LETTER B WITH LINE BELOW
<C,'>      <U1E08>   LATIN CAPITAL LETTER C WITH CEDILLA AND ACUTE
<c,'>      <U1E09>   LATIN SMALL LETTER C WITH CEDILLA AND ACUTE
<D.>       <U1E0A>   LATIN CAPITAL LETTER D WITH DOT ABOVE
<d.>       <U1E0B>   LATIN SMALL LETTER D WITH DOT ABOVE
<D-.>      <U1E0C>   LATIN CAPITAL LETTER D WITH DOT BELOW
<d-.>      <U1E0D>   LATIN SMALL LETTER D WITH DOT BELOW
<D_>       <U1E0E>   LATIN CAPITAL LETTER D WITH LINE BELOW
<d_>       <U1E0F>   LATIN SMALL LETTER D WITH LINE BELOW
<D,>       <U1E10>   LATIN CAPITAL LETTER D WITH CEDILLA
<d,>       <U1E11>   LATIN SMALL LETTER D WITH CEDILLA
<D-/>>     <U1E12>   LATIN CAPITAL LETTER D WITH CIRCUMFLEX BELOW
<d-/>>     <U1E13>   LATIN SMALL LETTER D WITH CIRCUMFLEX BELOW
<E-!>      <U1E14>   LATIN CAPITAL LETTER E WITH MACRON AND GRAVE
<e-!>      <U1E15>   LATIN SMALL LETTER E WITH MACRON AND GRAVE
<E-'>      <U1E16>   LATIN CAPITAL LETTER E WITH MACRON AND ACUTE
<e-'>      <U1E17>   LATIN SMALL LETTER E WITH MACRON AND ACUTE
<E-/>>     <U1E18>   LATIN CAPITAL LETTER E WITH CIRCUMFLEX BELOW
<e-/>>     <U1E19>   LATIN SMALL LETTER E WITH CIRCUMFLEX BELOW
<E-?>      <U1E1A>   LATIN CAPITAL LETTER E WITH TILDE BELOW
<e-?>      <U1E1B>   LATIN SMALL LETTER E WITH TILDE BELOW
<E,(>      <U1E1C>   LATIN CAPITAL LETTER E WITH CEDILLA AND BREVE
<e,(>      <U1E1D>   LATIN SMALL LETTER E WITH CEDILLA AND BREVE
<F.>       <U1E1E>   LATIN CAPITAL LETTER F WITH DOT ABOVE
<f.>       <U1E1F>   LATIN SMALL LETTER F WITH DOT ABOVE
<G->       <U1E20>   LATIN CAPITAL LETTER G WITH MACRON
<g->       <U1E21>   LATIN SMALL LETTER G WITH MACRON
<H.>       <U1E22>   LATIN CAPITAL LETTER H WITH DOT ABOVE
<h.>       <U1E23>   LATIN SMALL LETTER H WITH DOT ABOVE
<H-.>      <U1E24>   LATIN CAPITAL LETTER H WITH DOT BELOW
<h-.>      <U1E25>   LATIN SMALL LETTER H WITH DOT BELOW
<H:>       <U1E26>   LATIN CAPITAL LETTER H WITH DIAERESIS
<h:>       <U1E27>   LATIN SMALL LETTER H WITH DIAERESIS
<H,>       <U1E28>   LATIN CAPITAL LETTER H WITH CEDILLA
<h,>       <U1E29>   LATIN SMALL LETTER H WITH CEDILLA
<H-(>      <U1E2A>   LATIN CAPITAL LETTER H WITH BREVE BELOW
<h-(>      <U1E2B>   LATIN SMALL LETTER H WITH BREVE BELOW
```



```
<I-?>      <U1E2C>   LATIN CAPITAL LETTER I WITH TILDE BELOW
<i-?>      <U1E2D>   LATIN SMALL LETTER I WITH TILDE BELOW
<I:'>      <U1E2E>   LATIN CAPITAL LETTER I WITH DIAERESIS AND ACUTE
<i:'>      <U1E2F>   LATIN SMALL LETTER I WITH DIAERESIS AND ACUTE
<K'>       <U1E30>   LATIN CAPITAL LETTER K WITH ACUTE
<k'>       <U1E31>   LATIN SMALL LETTER K WITH ACUTE
<K-.>      <U1E32>   LATIN CAPITAL LETTER K WITH DOT BELOW
<k-.>      <U1E33>   LATIN SMALL LETTER K WITH DOT BELOW
<K_>       <U1E34>   LATIN CAPITAL LETTER K WITH LINE BELOW
<k_>       <U1E35>   LATIN SMALL LETTER K WITH LINE BELOW
<L-.>      <U1E36>   LATIN CAPITAL LETTER L WITH DOT BELOW
<l-.>      <U1E37>   LATIN SMALL LETTER L WITH DOT BELOW
<L--.>     <U1E38>   LATIN CAPITAL LETTER L WITH DOT BELOW AND MACRON
<l--.>     <U1E39>   LATIN SMALL LETTER L WITH DOT BELOW AND MACRON
<L_>       <U1E3A>   LATIN CAPITAL LETTER L WITH LINE BELOW
<l_>       <U1E3B>   LATIN SMALL LETTER L WITH LINE BELOW
<L-/>>     <U1E3C>   LATIN CAPITAL LETTER L WITH CIRCUMFLEX BELOW
<l-/>>     <U1E3D>   LATIN SMALL LETTER L WITH CIRCUMFLEX BELOW
<M'>       <U1E3E>   LATIN CAPITAL LETTER M WITH ACUTE
<m'>       <U1E3F>   LATIN SMALL LETTER M WITH ACUTE
<M.>       <U1E40>   LATIN CAPITAL LETTER M WITH DOT ABOVE
<m.>       <U1E41>   LATIN SMALL LETTER M WITH DOT ABOVE
<M-.>      <U1E42>   LATIN CAPITAL LETTER M WITH DOT BELOW
<m-.>      <U1E43>   LATIN SMALL LETTER M WITH DOT BELOW
<N.>       <U1E44>   LATIN CAPITAL LETTER N WITH DOT ABOVE
<n.>       <U1E45>   LATIN SMALL LETTER N WITH DOT ABOVE
<N-.>      <U1E46>   LATIN CAPITAL LETTER N WITH DOT BELOW
<n-.>      <U1E47>   LATIN SMALL LETTER N WITH DOT BELOW
<N_>       <U1E48>   LATIN CAPITAL LETTER N WITH LINE BELOW
<n_>       <U1E49>   LATIN SMALL LETTER N WITH LINE BELOW
<N-/>>     <U1E4A>   LATIN CAPITAL LETTER N WITH CIRCUMFLEX BELOW
<n-/>>     <U1E4B>   LATIN SMALL LETTER N WITH CIRCUMFLEX BELOW
<O?'>      <U1E4C>   LATIN CAPITAL LETTER O WITH TILDE AND ACUTE
<o?'>      <U1E4D>   LATIN SMALL LETTER O WITH TILDE AND ACUTE
<O?:>      <U1E4E>   LATIN CAPITAL LETTER O WITH TILDE AND DIAERESIS
<o?:>      <U1E4F>   LATIN SMALL LETTER O WITH TILDE AND DIAERESIS
<O-!>      <U1E50>   LATIN CAPITAL LETTER O WITH MACRON AND GRAVE
<o-!>      <U1E51>   LATIN SMALL LETTER O WITH MACRON AND GRAVE
<O-'>      <U1E52>   LATIN CAPITAL LETTER O WITH MACRON AND ACUTE
<o-'>      <U1E53>   LATIN SMALL LETTER O WITH MACRON AND ACUTE
<P'>       <U1E54>   LATIN CAPITAL LETTER P WITH ACUTE
<p'>       <U1E55>   LATIN SMALL LETTER P WITH ACUTE
<P.>       <U1E56>   LATIN CAPITAL LETTER P WITH DOT ABOVE
<p.>       <U1E57>   LATIN SMALL LETTER P WITH DOT ABOVE
<R.>       <U1E58>   LATIN CAPITAL LETTER R WITH DOT ABOVE
<r.>       <U1E59>   LATIN SMALL LETTER R WITH DOT ABOVE
<R-.>      <U1E5A>   LATIN CAPITAL LETTER R WITH DOT BELOW
<r-.>      <U1E5B>   LATIN SMALL LETTER R WITH DOT BELOW
<R--.>     <U1E5C>   LATIN CAPITAL LETTER R WITH DOT BELOW AND MACRON
<r--.>     <U1E5D>   LATIN SMALL LETTER R WITH DOT BELOW AND MACRON
<R_>       <U1E5E>   LATIN CAPITAL LETTER R WITH LINE BELOW
<r_>       <U1E5F>   LATIN SMALL LETTER R WITH LINE BELOW
<S.>       <U1E60>   LATIN CAPITAL LETTER S WITH DOT ABOVE
<s.>       <U1E61>   LATIN SMALL LETTER S WITH DOT ABOVE
<S-.>      <U1E62>   LATIN CAPITAL LETTER S WITH DOT BELOW
<s-.>      <U1E63>   LATIN SMALL LETTER S WITH DOT BELOW
<S'.>      <U1E64>   LATIN CAPITAL LETTER S WITH ACUTE AND DOT ABOVE
<s'.>      <U1E65>   LATIN SMALL LETTER S WITH ACUTE AND DOT ABOVE
<S<.>      <U1E66>   LATIN CAPITAL LETTER S WITH CARON AND DOT ABOVE
<s<.>      <U1E67>   LATIN SMALL LETTER S WITH CARON AND DOT ABOVE
<S.-.>     <U1E68>   LATIN CAPITAL LETTER S WITH DOT BELOW AND DOT ABOVE
<s.-.>     <U1E69>   LATIN SMALL LETTER S WITH DOT BELOW AND DOT ABOVE
<T.>       <U1E6A>   LATIN CAPITAL LETTER T WITH DOT ABOVE
<t.>       <U1E6B>   LATIN SMALL LETTER T WITH DOT ABOVE
<T-.>      <U1E6C>   LATIN CAPITAL LETTER T WITH DOT BELOW
<t-.>      <U1E6D>   LATIN SMALL LETTER T WITH DOT BELOW
<T_>       <U1E6E>   LATIN CAPITAL LETTER T WITH LINE BELOW
<t_>       <U1E6F>   LATIN SMALL LETTER T WITH LINE BELOW
<T-/>>     <U1E70>   LATIN CAPITAL LETTER T WITH CIRCUMFLEX BELOW
<t-/>>     <U1E71>   LATIN SMALL LETTER T WITH CIRCUMFLEX BELOW
<U--:>     <U1E72>   LATIN CAPITAL LETTER U WITH DIAERESIS BELOW
<u--:>     <U1E73>   LATIN SMALL LETTER U WITH DIAERESIS BELOW
<U-?>      <U1E74>   LATIN CAPITAL LETTER U WITH TILDE BELOW
<u-?>      <U1E75>   LATIN SMALL LETTER U WITH TILDE BELOW
```



```
<U-/>>     <U1E76>   LATIN CAPITAL LETTER U WITH CIRCUMFLEX BELOW
<u-/>>     <U1E77>   LATIN SMALL LETTER U WITH CIRCUMFLEX BELOW
<U?'>      <U1E78>   LATIN CAPITAL LETTER U WITH TILDE AND ACUTE
<u?'>      <U1E79>   LATIN SMALL LETTER U WITH TILDE AND ACUTE
<U-:>      <U1E7A>   LATIN CAPITAL LETTER U WITH MACRON AND DIAERESIS
<u-:>      <U1E7B>   LATIN SMALL LETTER U WITH MACRON AND DIAERESIS
<V?>       <U1E7C>   LATIN CAPITAL LETTER V WITH TILDE
<v?>       <U1E7D>   LATIN SMALL LETTER V WITH TILDE
<V-.>      <U1E7E>   LATIN CAPITAL LETTER V WITH DOT BELOW
<v-.>      <U1E7F>   LATIN SMALL LETTER V WITH DOT BELOW
<W!>       <U1E80>   LATIN CAPITAL LETTER W WITH GRAVE
<w!>       <U1E81>   LATIN SMALL LETTER W WITH GRAVE
<W'>       <U1E82>   LATIN CAPITAL LETTER W WITH ACUTE
<w'>       <U1E83>   LATIN SMALL LETTER W WITH ACUTE
<W:>       <U1E84>   LATIN CAPITAL LETTER W WITH DIAERESIS
<w:>       <U1E85>   LATIN SMALL LETTER W WITH DIAERESIS
<W.>       <U1E86>   LATIN CAPITAL LETTER W WITH DOT ABOVE
<w.>       <U1E87>   LATIN SMALL LETTER W WITH DOT ABOVE
<W-.>      <U1E88>   LATIN CAPITAL LETTER W WITH DOT BELOW
<w-.>      <U1E89>   LATIN SMALL LETTER W WITH DOT BELOW
<X.>       <U1E8A>   LATIN CAPITAL LETTER X WITH DOT ABOVE
<x.>       <U1E8B>   LATIN SMALL LETTER X WITH DOT ABOVE
<X:>       <U1E8C>   LATIN CAPITAL LETTER X WITH DIAERESIS
<x:>       <U1E8D>   LATIN SMALL LETTER X WITH DIAERESIS
<Y.>       <U1E8E>   LATIN CAPITAL LETTER Y WITH DOT ABOVE
<y.>       <U1E8F>   LATIN SMALL LETTER Y WITH DOT ABOVE
<Z/>>      <U1E90>   LATIN CAPITAL LETTER Z WITH CIRCUMFLEX
<z/>>      <U1E91>   LATIN SMALL LETTER Z WITH CIRCUMFLEX
<Z-.>      <U1E92>   LATIN CAPITAL LETTER Z WITH DOT BELOW
<z-.>      <U1E93>   LATIN SMALL LETTER Z WITH DOT BELOW
<Z_>       <U1E94>   LATIN CAPITAL LETTER Z WITH LINE BELOW
<z_>       <U1E95>   LATIN SMALL LETTER Z WITH LINE BELOW
<A-.>      <U1EA0>   LATIN CAPITAL LETTER A WITH DOT BELOW
<a-.>      <U1EA1>   LATIN SMALL LETTER A WITH DOT BELOW
<A2>       <U1EA2>   LATIN CAPITAL LETTER A WITH HOOK ABOVE
<a2>       <U1EA3>   LATIN SMALL LETTER A WITH HOOK ABOVE
<A/>'>     <U1EA4>   LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND ACUTE
<a/>'>     <U1EA5>   LATIN SMALL LETTER A WITH CIRCUMFLEX AND ACUTE
<A/>!>     <U1EA6>   LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND GRAVE
<a/>!>     <U1EA7>   LATIN SMALL LETTER A WITH CIRCUMFLEX AND GRAVE
<A/>2>     <U1EA8>   LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND HOOK ABOVE
<a/>2>     <U1EA9>   LATIN SMALL LETTER A WITH CIRCUMFLEX AND HOOK ABOVE
<A/>?>     <U1EAA>   LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND TILDE
<a/>?>     <U1EAB>   LATIN SMALL LETTER A WITH CIRCUMFLEX AND TILDE
<A/>-.>    <U1EAC>   LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND DOT BELOW
<a/>-.>    <U1EAD>   LATIN SMALL LETTER A WITH CIRCUMFLEX AND DOT BELOW
<A('>      <U1EAE>   LATIN CAPITAL LETTER A WITH BREVE AND ACUTE
<a('>      <U1EAF>   LATIN SMALL LETTER A WITH BREVE AND ACUTE
<A(!>      <U1EB0>   LATIN CAPITAL LETTER A WITH BREVE AND GRAVE
<a(!>      <U1EB1>   LATIN SMALL LETTER A WITH BREVE AND GRAVE
<A(2>      <U1EB2>   LATIN CAPITAL LETTER A WITH BREVE AND HOOK ABOVE
<a(2>      <U1EB3>   LATIN SMALL LETTER A WITH BREVE AND HOOK ABOVE
<A(?>      <U1EB4>   LATIN CAPITAL LETTER A WITH BREVE AND TILDE
<a(?>      <U1EB5>   LATIN SMALL LETTER A WITH BREVE AND TILDE
<A(-.>     <U1EB6>   LATIN CAPITAL LETTER A WITH BREVE AND DOT BELOW
<a(-.>     <U1EB7>   LATIN SMALL LETTER A WITH BREVE AND DOT BELOW
<E-.>      <U1EB8>   LATIN CAPITAL LETTER E WITH DOT BELOW
<e-.>      <U1EB9>   LATIN SMALL LETTER E WITH DOT BELOW
<E2>       <U1EBA>   LATIN CAPITAL LETTER E WITH HOOK ABOVE
<e2>       <U1EBB>   LATIN SMALL LETTER E WITH HOOK ABOVE
<E?>       <U1EBC>   LATIN CAPITAL LETTER E WITH TILDE
<e?>       <U1EBD>   LATIN SMALL LETTER E WITH TILDE
<E/>'>     <U1EBE>   LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND ACUTE
<e/>'>     <U1EBF>   LATIN SMALL LETTER E WITH CIRCUMFLEX AND ACUTE
<E/>!>     <U1EC0>   LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND GRAVE
<e/>!>     <U1EC1>   LATIN SMALL LETTER E WITH CIRCUMFLEX AND GRAVE
<E/>2>     <U1EC2>   LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE
<e/>2>     <U1EC3>   LATIN SMALL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE
<E/>?>     <U1EC4>   LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND TILDE
<e/>?>     <U1EC5>   LATIN SMALL LETTER E WITH CIRCUMFLEX AND TILDE
<E/>-.>    <U1EC6>   LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND DOT BELOW
<e/>-.>    <U1EC7>   LATIN SMALL LETTER E WITH CIRCUMFLEX AND DOT BELOW
<I2>       <U1EC8>   LATIN CAPITAL LETTER I WITH HOOK ABOVE
<i2>       <U1EC9>   LATIN SMALL LETTER I WITH HOOK ABOVE
```



```
<I-.>      <U1ECA>   LATIN CAPITAL LETTER I WITH DOT BELOW
<i-.>      <U1ECB>   LATIN SMALL LETTER I WITH DOT BELOW
<O-.>      <U1ECC>   LATIN CAPITAL LETTER O WITH DOT BELOW
<o-.>      <U1ECD>   LATIN SMALL LETTER O WITH DOT BELOW
<O2>       <U1ECE>   LATIN CAPITAL LETTER O WITH HOOK ABOVE
<o2>       <U1ECF>   LATIN SMALL LETTER O WITH HOOK ABOVE
<O/>'>     <U1ED0>   LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND ACUTE
<o/>'>     <U1ED1>   LATIN SMALL LETTER O WITH CIRCUMFLEX AND ACUTE
<O/>!>     <U1ED2>   LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND GRAVE
<o/>!>     <U1ED3>   LATIN SMALL LETTER O WITH CIRCUMFLEX AND GRAVE
<O/>2>     <U1ED4>   LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE
<o/>2>     <U1ED5>   LATIN SMALL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE
<O/>?>     <U1ED6>   LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND TILDE
<o/>?>     <U1ED7>   LATIN SMALL LETTER O WITH CIRCUMFLEX AND TILDE
<O/>-.>    <U1ED8>   LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND DOT BELOW
<o/>-.>    <U1ED9>   LATIN SMALL LETTER O WITH CIRCUMFLEX AND DOT BELOW
<O9'>      <U1EDA>   LATIN CAPITAL LETTER O WITH HORN AND ACUTE
<o9'>      <U1EDB>   LATIN SMALL LETTER O WITH HORN AND ACUTE
<O9!>      <U1EDC>   LATIN CAPITAL LETTER O WITH HORN AND GRAVE
<o9!>      <U1EDD>   LATIN SMALL LETTER O WITH HORN AND GRAVE
<O92>      <U1EDE>   LATIN CAPITAL LETTER O WITH HORN AND HOOK ABOVE
<o92>      <U1EDF>   LATIN SMALL LETTER O WITH HORN AND HOOK ABOVE
<O9?>      <U1EE0>   LATIN CAPITAL LETTER O WITH HORN AND TILDE
<o9?>      <U1EE1>   LATIN SMALL LETTER O WITH HORN AND TILDE
<O9-.>     <U1EE2>   LATIN CAPITAL LETTER O WITH HORN AND DOT BELOW
<o9-.>     <U1EE3>   LATIN SMALL LETTER O WITH HORN AND DOT BELOW
<U-.>      <U1EE4>   LATIN CAPITAL LETTER U WITH DOT BELOW
<u-.>      <U1EE5>   LATIN SMALL LETTER U WITH DOT BELOW
<U2>       <U1EE6>   LATIN CAPITAL LETTER U WITH HOOK ABOVE
<u2>       <U1EE7>   LATIN SMALL LETTER U WITH HOOK ABOVE
<U9'>      <U1EE8>   LATIN CAPITAL LETTER U WITH HORN AND ACUTE
<u9'>      <U1EE9>   LATIN SMALL LETTER U WITH HORN AND ACUTE
<U9!>      <U1EEA>   LATIN CAPITAL LETTER U WITH HORN AND GRAVE
<u9!>      <U1EEB>   LATIN SMALL LETTER U WITH HORN AND GRAVE
<U92>      <U1EEC>   LATIN CAPITAL LETTER U WITH HORN AND HOOK ABOVE
<u92>      <U1EED>   LATIN SMALL LETTER U WITH HORN AND HOOK ABOVE
<U9?>      <U1EEE>   LATIN CAPITAL LETTER U WITH HORN AND TILDE
<u9?>      <U1EEF>   LATIN SMALL LETTER U WITH HORN AND TILDE
<U9-.>     <U1EF0>   LATIN CAPITAL LETTER U WITH HORN AND DOT BELOW
<u9-.>     <U1EF1>   LATIN SMALL LETTER U WITH HORN AND DOT BELOW
<Y!>       <U1EF2>   LATIN CAPITAL LETTER Y WITH GRAVE
<y!>       <U1EF3>   LATIN SMALL LETTER Y WITH GRAVE
<Y-.>      <U1EF4>   LATIN CAPITAL LETTER Y WITH DOT BELOW
<y-.>      <U1EF5>   LATIN SMALL LETTER Y WITH DOT BELOW
<Y2>       <U1EF6>   LATIN CAPITAL LETTER Y WITH HOOK ABOVE
<y2>       <U1EF7>   LATIN SMALL LETTER Y WITH HOOK ABOVE
<Y?>       <U1EF8>   LATIN CAPITAL LETTER Y WITH TILDE
<y?>       <U1EF9>   LATIN SMALL LETTER Y WITH TILDE
<a*,>      <U1F00>   GREEK SMALL LETTER ALPHA WITH PSILI
<a*;>      <U1F01>   GREEK SMALL LETTER ALPHA WITH DASIA
<a*,!>     <U1F02>   GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA
<a*;!>     <U1F03>   GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA
<a*,'>     <U1F04>   GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA
<a*;'>     <U1F05>   GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA
<a*,?>     <U1F06>   GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI
<a*;?>     <U1F07>   GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI
<A*,>      <U1F08>   GREEK CAPITAL LETTER ALPHA WITH PSILI
<A*;>      <U1F09>   GREEK CAPITAL LETTER ALPHA WITH DASIA
<A*,!>     <U1F0A>   GREEK CAPITAL LETTER ALPHA WITH PSILI AND VARIA
<A*;!>     <U1F0B>   GREEK CAPITAL LETTER ALPHA WITH DASIA AND VARIA
<A*,'>     <U1F0C>   GREEK CAPITAL LETTER ALPHA WITH PSILI AND OXIA
<A*;'>     <U1F0D>   GREEK CAPITAL LETTER ALPHA WITH DASIA AND OXIA
<A*,?>     <U1F0E>   GREEK CAPITAL LETTER ALPHA WITH PSILI AND PERISPOMENI
<A*;?>     <U1F0F>   GREEK CAPITAL LETTER ALPHA WITH DASIA AND PERISPOMENI
<e*,>      <U1F10>   GREEK SMALL LETTER EPSILON WITH PSILI
<e*;>      <U1F11>   GREEK SMALL LETTER EPSILON WITH DASIA
<e*,!>     <U1F12>   GREEK SMALL LETTER EPSILON WITH PSILI AND VARIA
<e*;!>     <U1F13>   GREEK SMALL LETTER EPSILON WITH DASIA AND VARIA
<e*,'>     <U1F14>   GREEK SMALL LETTER EPSILON WITH PSILI AND OXIA
<e*;'>     <U1F15>   GREEK SMALL LETTER EPSILON WITH DASIA AND OXIA
<E*,>      <U1F18>   GREEK CAPITAL LETTER EPSILON WITH PSILI
<E*;>      <U1F19>   GREEK CAPITAL LETTER EPSILON WITH DASIA
<E*,!>     <U1F1A>   GREEK CAPITAL LETTER EPSILON WITH PSILI AND VARIA
<E*;!>     <U1F1B>   GREEK CAPITAL LETTER EPSILON WITH DASIA AND VARIA
```



```
<E*,'>     <U1F1C>   GREEK CAPITAL LETTER EPSILON WITH PSILI AND OXIA
<E*;'>     <U1F1D>   GREEK CAPITAL LETTER EPSILON WITH DASIA AND OXIA
<y*,>      <U1F20>   GREEK SMALL LETTER ETA WITH PSILI
<y*;>      <U1F21>   GREEK SMALL LETTER ETA WITH DASIA
<y*,!>     <U1F22>   GREEK SMALL LETTER ETA WITH PSILI AND VARIA
<y*;!>     <U1F23>   GREEK SMALL LETTER ETA WITH DASIA AND VARIA
<y*,'>     <U1F24>   GREEK SMALL LETTER ETA WITH PSILI AND OXIA
<y*;'>     <U1F25>   GREEK SMALL LETTER ETA WITH DASIA AND OXIA
<y*,?>     <U1F26>   GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI
<y*;?>     <U1F27>   GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI
<Y*,>      <U1F28>   GREEK CAPITAL LETTER ETA WITH PSILI
<Y*;>      <U1F29>   GREEK CAPITAL LETTER ETA WITH DASIA
<Y*,!>     <U1F2A>   GREEK CAPITAL LETTER ETA WITH PSILI AND VARIA
<Y*;!>     <U1F2B>   GREEK CAPITAL LETTER ETA WITH DASIA AND VARIA
<Y*,'>     <U1F2C>   GREEK CAPITAL LETTER ETA WITH PSILI AND OXIA
<Y*;'>     <U1F2D>   GREEK CAPITAL LETTER ETA WITH DASIA AND OXIA
<Y*,?>     <U1F2E>   GREEK CAPITAL LETTER ETA WITH PSILI AND PERISPOMENI
<Y*;?>     <U1F2F>   GREEK CAPITAL LETTER ETA WITH DASIA AND PERISPOMENI
<i*,>      <U1F30>   GREEK SMALL LETTER IOTA WITH PSILI
<i*;>      <U1F31>   GREEK SMALL LETTER IOTA WITH DASIA
<i*,!>     <U1F32>   GREEK SMALL LETTER IOTA WITH PSILI AND VARIA
<i*;!>     <U1F33>   GREEK SMALL LETTER IOTA WITH DASIA AND VARIA
<i*,'>     <U1F34>   GREEK SMALL LETTER IOTA WITH PSILI AND OXIA
<i*;'>     <U1F35>   GREEK SMALL LETTER IOTA WITH DASIA AND OXIA
<i*,?>     <U1F36>   GREEK SMALL LETTER IOTA WITH PSILI AND PERISPOMENI
<i*;?>     <U1F37>   GREEK SMALL LETTER IOTA WITH DASIA AND PERISPOMENI
<I*,>      <U1F38>   GREEK CAPITAL LETTER IOTA WITH PSILI
<I*;>      <U1F39>   GREEK CAPITAL LETTER IOTA WITH DASIA
<I*,!>     <U1F3A>   GREEK CAPITAL LETTER IOTA WITH PSILI AND VARIA
<I*;!>     <U1F3B>   GREEK CAPITAL LETTER IOTA WITH DASIA AND VARIA
<I*,'>     <U1F3C>   GREEK CAPITAL LETTER IOTA WITH PSILI AND OXIA
<I*;'>     <U1F3D>   GREEK CAPITAL LETTER IOTA WITH DASIA AND OXIA
<I*,?>     <U1F3E>   GREEK CAPITAL LETTER IOTA WITH PSILI AND PERISPOMENI
<I*;?>     <U1F3F>   GREEK CAPITAL LETTER IOTA WITH DASIA AND PERISPOMENI
<o*,>      <U1F40>   GREEK SMALL LETTER OMICRON WITH PSILI
<o*;>      <U1F41>   GREEK SMALL LETTER OMICRON WITH DASIA
<o*,!>     <U1F42>   GREEK SMALL LETTER OMICRON WITH PSILI AND VARIA
<o*;!>     <U1F43>   GREEK SMALL LETTER OMICRON WITH DASIA AND VARIA
<o*,'>     <U1F44>   GREEK SMALL LETTER OMICRON WITH PSILI AND OXIA
<o*;'>     <U1F45>   GREEK SMALL LETTER OMICRON WITH DASIA AND OXIA
<O*,>      <U1F48>   GREEK CAPITAL LETTER OMICRON WITH PSILI
<O*;>      <U1F49>   GREEK CAPITAL LETTER OMICRON WITH DASIA
<O*,!>     <U1F4A>   GREEK CAPITAL LETTER OMICRON WITH PSILI AND VARIA
<O*;!>     <U1F4B>   GREEK CAPITAL LETTER OMICRON WITH DASIA AND VARIA
<O*,'>     <U1F4C>   GREEK CAPITAL LETTER OMICRON WITH PSILI AND OXIA
<O*;'>     <U1F4D>   GREEK CAPITAL LETTER OMICRON WITH DASIA AND OXIA
<u*,>      <U1F50>   GREEK SMALL LETTER UPSILON WITH PSILI
<u*;>      <U1F51>   GREEK SMALL LETTER UPSILON WITH DASIA
<u*,!>     <U1F52>   GREEK SMALL LETTER UPSILON WITH PSILI AND VARIA
<u*;!>     <U1F53>   GREEK SMALL LETTER UPSILON WITH DASIA AND VARIA
<u*,'>     <U1F54>   GREEK SMALL LETTER UPSILON WITH PSILI AND OXIA
<u*;'>     <U1F55>   GREEK SMALL LETTER UPSILON WITH DASIA AND OXIA
<u*,?>     <U1F56>   GREEK SMALL LETTER UPSILON WITH PSILI AND PERISPOMENI
<u*;?>     <U1F57>   GREEK SMALL LETTER UPSILON WITH DASIA AND PERISPOMENI
<U*;>      <U1F59>   GREEK CAPITAL LETTER UPSILON WITH DASIA
<U*;!>     <U1F5B>   GREEK CAPITAL LETTER UPSILON WITH DASIA AND VARIA
<U*;'>     <U1F5D>   GREEK CAPITAL LETTER UPSILON WITH DASIA AND OXIA
<U*;?>     <U1F5F>   GREEK CAPITAL LETTER UPSILON WITH DASIA AND PERISPOMENI
<w*,>      <U1F60>   GREEK SMALL LETTER OMEGA WITH PSILI
<w*;>      <U1F61>   GREEK SMALL LETTER OMEGA WITH DASIA
<w*,!>     <U1F62>   GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA
<w*;!>     <U1F63>   GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA
<w*,'>     <U1F64>   GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA
<w*;'>     <U1F65>   GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA
<w*,?>     <U1F66>   GREEK SMALL LETTER OMEGA WITH PSILI AND PERISPOMENI
<w*;?>     <U1F67>   GREEK SMALL LETTER OMEGA WITH DASIA AND PERISPOMENI
<W*,>      <U1F68>   GREEK CAPITAL LETTER OMEGA WITH PSILI
<W*;>      <U1F69>   GREEK CAPITAL LETTER OMEGA WITH DASIA
<W*,!>     <U1F6A>   GREEK CAPITAL LETTER OMEGA WITH PSILI AND VARIA
<W*;!>     <U1F6B>   GREEK CAPITAL LETTER OMEGA WITH DASIA AND VARIA
<W*,'>     <U1F6C>   GREEK CAPITAL LETTER OMEGA WITH PSILI AND OXIA
<W*;'>     <U1F6D>   GREEK CAPITAL LETTER OMEGA WITH DASIA AND OXIA
<W*,?>     <U1F6E>   GREEK CAPITAL LETTER OMEGA WITH PSILI AND PERISPOMENI
<W*;?>     <U1F6F>   GREEK CAPITAL LETTER OMEGA WITH DASIA AND PERISPOMENI
```



```
<a*!>      <U1F70>   GREEK SMALL LETTER ALPHA WITH VARIA
<a*'>      <U1F71>   GREEK SMALL LETTER ALPHA WITH OXIA
<e*!>      <U1F72>   GREEK SMALL LETTER EPSILON WITH VARIA
<e*'>      <U1F73>   GREEK SMALL LETTER EPSILON WITH OXIA
<y*!>      <U1F74>   GREEK SMALL LETTER ETA WITH VARIA
<y*'>      <U1F75>   GREEK SMALL LETTER ETA WITH OXIA
<i*!>      <U1F76>   GREEK SMALL LETTER IOTA WITH VARIA
<i*'>      <U1F77>   GREEK SMALL LETTER IOTA WITH OXIA
<o*!>      <U1F78>   GREEK SMALL LETTER OMICRON WITH VARIA
<o*'>      <U1F79>   GREEK SMALL LETTER OMICRON WITH OXIA
<u*!>      <U1F7A>   GREEK SMALL LETTER UPSILON WITH VARIA
<u*'>      <U1F7B>   GREEK SMALL LETTER UPSILON WITH OXIA
<w*!>      <U1F7C>   GREEK SMALL LETTER OMEGA WITH VARIA
<w*'>      <U1F7D>   GREEK SMALL LETTER OMEGA WITH OXIA
<a*,j>     <U1F80>   GREEK SMALL LETTER ALPHA WITH PSILI AND YPOGEGRAMMENI
<a*;j>     <U1F81>   GREEK SMALL LETTER ALPHA WITH DASIA AND YPOGEGRAMMENI
<a*,!j>    <U1F82>   GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA AND YPOGEGRAMMENI
<a*;!j>    <U1F83>   GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA AND YPOGEGRAMMENI
<a*,'j>    <U1F84>   GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA AND YPOGEGRAMMENI
<a*;'j>    <U1F85>   GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA AND YPOGEGRAMMENI
<a*,?j>    <U1F86>   GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
<a*;?j>    <U1F87>   GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
<A*,J>     <U1F88>   GREEK CAPITAL LETTER ALPHA WITH PSILI AND PROSGEGRAMMENI
<A*;J>     <U1F89>   GREEK CAPITAL LETTER ALPHA WITH DASIA AND PROSGEGRAMMENI
<A*,!J>    <U1F8A>   GREEK CAPITAL LETTER ALPHA WITH PSILI AND VARIA AND PROSGEGRAMMENI
<A*;!J>    <U1F8B>   GREEK CAPITAL LETTER ALPHA WITH DASIA AND VARIA AND PROSGEGRAMMENI
<A*,'J>    <U1F8C>   GREEK CAPITAL LETTER ALPHA WITH PSILI AND OXIA AND PROSGEGRAMMENI
<A*;'J>    <U1F8D>   GREEK CAPITAL LETTER ALPHA WITH DASIA AND OXIA AND PROSGEGRAMMENI
<A*,?J>    <U1F8E>   GREEK CAPITAL LETTER ALPHA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
<A*;?J>    <U1F8F>   GREEK CAPITAL LETTER ALPHA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI
<y*,j>     <U1F90>   GREEK SMALL LETTER ETA WITH PSILI AND YPOGEGRAMMENI
<y*;j>     <U1F91>   GREEK SMALL LETTER ETA WITH DASIA AND YPOGEGRAMMENI
<y*,!j>    <U1F92>   GREEK SMALL LETTER ETA WITH PSILI AND VARIA AND YPOGEGRAMMENI
<y*;!j>    <U1F93>   GREEK SMALL LETTER ETA WITH DASIA AND VARIA AND YPOGEGRAMMENI
<y*,'j>    <U1F94>   GREEK SMALL LETTER ETA WITH PSILI AND OXIA AND YPOGEGRAMMENI
<y*;'j>    <U1F95>   GREEK SMALL LETTER ETA WITH DASIA AND OXIA AND YPOGEGRAMMENI
<y*,?j>    <U1F96>   GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
<y*;?j>    <U1F97>   GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
<Y*,J>     <U1F98>   GREEK CAPITAL LETTER ETA WITH PSILI AND PROSGEGRAMMENI
<Y*;J>     <U1F99>   GREEK CAPITAL LETTER ETA WITH DASIA AND PROSGEGRAMMENI
<Y*,!J>    <U1F9A>   GREEK CAPITAL LETTER ETA WITH PSILI AND VARIA AND PROSGEGRAMMENI
<Y*;!J>    <U1F9B>   GREEK CAPITAL LETTER ETA WITH DASIA AND VARIA AND PROSGEGRAMMENI
<Y*,'J>    <U1F9C>   GREEK CAPITAL LETTER ETA WITH PSILI AND OXIA AND PROSGEGRAMMENI
<Y*;'J>    <U1F9D>   GREEK CAPITAL LETTER ETA WITH DASIA AND OXIA AND PROSGEGRAMMENI
<Y*,?J>    <U1F9E>   GREEK CAPITAL LETTER ETA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
<Y*;?J>    <U1F9F>   GREEK CAPITAL LETTER ETA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI
<w*,j>     <U1FA0>   GREEK SMALL LETTER OMEGA WITH PSILI AND YPOGEGRAMMENI
<w*;j>     <U1FA1>   GREEK SMALL LETTER OMEGA WITH DASIA AND YPOGEGRAMMENI
<w*,!j>    <U1FA2>   GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA AND YPOGEGRAMMENI
<w*;!j>    <U1FA3>   GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA AND YPOGEGRAMMENI
<w*,'j>    <U1FA4>   GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA AND YPOGEGRAMMENI
<w*;'j>    <U1FA5>   GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA AND YPOGEGRAMMENI
<w*,?j>    <U1FA6>   GREEK SMALL LETTER OMEGA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
<w*;?j>    <U1FA7>   GREEK SMALL LETTER OMEGA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
<W*,J>     <U1FA8>   GREEK CAPITAL LETTER OMEGA WITH PSILI AND PROSGEGRAMMENI
<W*;J>     <U1FA9>   GREEK CAPITAL LETTER OMEGA WITH DASIA AND PROSGEGRAMMENI
<W*,!J>    <U1FAA>   GREEK CAPITAL LETTER OMEGA WITH PSILI AND VARIA AND PROSGEGRAMMENI
<W*;!J>    <U1FAB>   GREEK CAPITAL LETTER OMEGA WITH DASIA AND VARIA AND PROSGEGRAMMENI
<W*,'J>    <U1FAC>   GREEK CAPITAL LETTER OMEGA WITH PSILI AND OXIA AND PROSGEGRAMMENI
<W*;'J>    <U1FAD>   GREEK CAPITAL LETTER OMEGA WITH DASIA AND OXIA AND PROSGEGRAMMENI
<W*,?J>    <U1FAE>   GREEK CAPITAL LETTER OMEGA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
<W*;?J>    <U1FAF>   GREEK CAPITAL LETTER OMEGA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI
<a*(>      <U1FB0>   GREEK SMALL LETTER ALPHA WITH VRACHY
<a*->      <U1FB1>   GREEK SMALL LETTER ALPHA WITH MACRON
<a*!j>     <U1FB2>   GREEK SMALL LETTER ALPHA WITH VARIA AND YPOGEGRAMMENI
<a*j>      <U1FB3>   GREEK SMALL LETTER ALPHA WITH YPOGEGRAMMENI
<a*'j>     <U1FB4>   GREEK SMALL LETTER ALPHA WITH OXIA AND YPOGEGRAMMENI
<a*?>      <U1FB6>   GREEK SMALL LETTER ALPHA WITH PERISPOMENI
<a*?j>     <U1FB7>   GREEK SMALL LETTER ALPHA WITH PERISPOMENI AND YPOGEGRAMMENI
<A*(>      <U1FB8>   GREEK CAPITAL LETTER ALPHA WITH VRACHY
<A*->      <U1FB9>   GREEK CAPITAL LETTER ALPHA WITH MACRON
<A*!>      <U1FBA>   GREEK CAPITAL LETTER ALPHA WITH VARIA
<A*'>      <U1FBB>   GREEK CAPITAL LETTER ALPHA WITH OXIA
<A*J>      <U1FBC>   GREEK CAPITAL LETTER ALPHA WITH PROSGEGRAMMENI
```



```
<)*>       <U1FBD>   GREEK KORONIS
<J3>       <U1FBE>   GREEK PROSGEGRAMMENI
<,,>       <U1FBF>   GREEK PSILI
<?*>       <U1FC0>   GREEK PERISPOMENI
<?:>       <U1FC1>   GREEK DIALYTIKA AND PERISPOMENI
<y*!j>     <U1FC2>   GREEK SMALL LETTER ETA WITH VARIA AND YPOGEGRAMMENI
<y*j>      <U1FC3>   GREEK SMALL LETTER ETA WITH YPOGEGRAMMENI
<y*'j>     <U1FC4>   GREEK SMALL LETTER ETA WITH OXIA AND YPOGEGRAMMENI
<y*?>      <U1FC6>   GREEK SMALL LETTER ETA WITH PERISPOMENI
<y*?j>     <U1FC7>   GREEK SMALL LETTER ETA WITH PERISPOMENI AND YPOGEGRAMMENI
<E*!!>     <U1FC8>   GREEK CAPITAL LETTER EPSILON WITH VARIA
<E*'>      <U1FC9>   GREEK CAPITAL LETTER EPSILON WITH OXIA
<Y*!>      <U1FCA>   GREEK CAPITAL LETTER ETA WITH VARIA
<Y*'>      <U1FCB>   GREEK CAPITAL LETTER ETA WITH OXIA
<Y*J>      <U1FCC>   GREEK CAPITAL LETTER ETA WITH PROSGEGRAMMENI
<,!>       <U1FCD>   GREEK PSILI AND VARIA
<,'>       <U1FCE>   GREEK PSILI AND OXIA
<?,>       <U1FCF>   GREEK PSILI AND PERISPOMENI
<i*(>      <U1FD0>   GREEK SMALL LETTER IOTA WITH VRACHY
<i*->      <U1FD1>   GREEK SMALL LETTER IOTA WITH MACRON
<i*:!>     <U1FD2>   GREEK SMALL LETTER IOTA WITH DIALYTIKA AND VARIA
<i*:'>     <U1FD3>   GREEK SMALL LETTER IOTA WITH DIALYTIKA AND OXIA
<i*?>      <U1FD6>   GREEK SMALL LETTER IOTA WITH PERISPOMENI
<i*:?>     <U1FD7>   GREEK SMALL LETTER IOTA WITH DIALYTIKA AND PERISPOMENI
<I*(>      <U1FD8>   GREEK CAPITAL LETTER IOTA WITH VRACHY
<I*->      <U1FD9>   GREEK CAPITAL LETTER IOTA WITH MACRON
<I*!>      <U1FDA>   GREEK CAPITAL LETTER IOTA WITH VARIA
<I*'>      <U1FDB>   GREEK CAPITAL LETTER IOTA WITH OXIA
<;!>       <U1FDD>   GREEK DASIA AND VARIA
<;'>       <U1FDE>   GREEK DASIA AND OXIA
<?;>       <U1FDF>   GREEK DASIA AND PERISPOMENI
<u*(>      <U1FE0>   GREEK SMALL LETTER UPSILON WITH VRACHY
<u*->      <U1FE1>   GREEK SMALL LETTER UPSILON WITH MACRON
<u*:!>     <U1FE2>   GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND VARIA
<u*:'>     <U1FE3>   GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND OXIA
<r*,>      <U1FE4>   GREEK SMALL LETTER RHO WITH PSILI
<r*;>      <U1FE5>   GREEK SMALL LETTER RHO WITH DASIA
<u*?>      <U1FE6>   GREEK SMALL LETTER UPSILON WITH PERISPOMENI
<u*:?>     <U1FE7>   GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND PERISPOMENI
<U*(>      <U1FE8>   GREEK CAPITAL LETTER UPSILON WITH VRACHY
<U*->      <U1FE9>   GREEK CAPITAL LETTER UPSILON WITH MACRON
<U*!>      <U1FEA>   GREEK CAPITAL LETTER UPSILON WITH VARIA
<U*'>      <U1FEB>   GREEK CAPITAL LETTER UPSILON WITH OXIA
<R*;>      <U1FEC>   GREEK CAPITAL LETTER RHO WITH DASIA
<!:>       <U1FED>   GREEK DIALYTIKA AND VARIA
<:'>       <U1FEE>   GREEK DIALYTIKA AND OXIA
<!*>       <U1FEF>   GREEK VARIA
<w*!j>     <U1FF2>   GREEK SMALL LETTER OMEGA WITH VARIA AND YPOGEGRAMMENI
<w*j>      <U1FF3>   GREEK SMALL LETTER OMEGA WITH YPOGEGRAMMENI
<w*'j>     <U1FF4>   GREEK SMALL LETTER OMEGA WITH OXIA AND YPOGEGRAMMENI
<w*?>      <U1FF6>   GREEK SMALL LETTER OMEGA WITH PERISPOMENI
<w*?j>     <U1FF7>   GREEK SMALL LETTER OMEGA WITH PERISPOMENI AND YPOGEGRAMMENI
<O*!>      <U1FF8>   GREEK CAPITAL LETTER OMICRON WITH VARIA
<O*'>      <U1FF9>   GREEK CAPITAL LETTER OMICRON WITH OXIA
<W*!>      <U1FFA>   GREEK CAPITAL LETTER OMEGA WITH VARIA
<W*'>      <U1FFB>   GREEK CAPITAL LETTER OMEGA WITH OXIA
<W*J>      <U1FFC>   GREEK CAPITAL LETTER OMEGA WITH PROSGEGRAMMENI
<//*>      <U1FFD>   GREEK OXIA
<;;>       <U1FFE>   GREEK DASIA
<1N>       <U2002>   EN SPACE
<1M>       <U2003>   EM SPACE
<3M>       <U2004>   THREE-PER-EM SPACE
<4M>       <U2005>   FOUR-PER-EM SPACE
<6M>       <U2006>   SIX-PER-EM SPACE
<LR>       <U200E>   LEFT-TO-RIGHT MARK
<RL>       <U200F>   RIGHT-TO-LEFT MARK
<1T>       <U2009>   THIN SPACE
<1H>       <U200A>   HAIR SPACE
<-1>       <U2010>   HYPHEN
<-N>       <U2013>   EN DASH
<-M>       <U2014>   EM DASH
<-3>       <U2015>   HORIZONTAL BAR
<!2>       <U2016>   DOUBLE VERTICAL LINE
<=2>       <U2017>   DOUBLE LOW LINE
```



```
<'6>       <U2018>   LEFT SINGLE QUOTATION MARK
<'9>       <U2019>   RIGHT SINGLE QUOTATION MARK
<.9>       <U201A>   SINGLE LOW-9 QUOTATION MARK
<9'>       <U201B>   SINGLE HIGH-REVERSED-9 QUOTATION MARK
<"6>       <U201C>   LEFT DOUBLE QUOTATION MARK
<"9>       <U201D>   RIGHT DOUBLE QUOTATION MARK
<:9>       <U201E>   DOUBLE LOW-9 QUOTATION MARK
<9">       <U201F>   DOUBLE HIGH-REVERSED-9 QUOTATION MARK
<//->      <U2020>   DAGGER
<//=>      <U2021>   DOUBLE DAGGER
<sb>       <U2022>   BULLET
<3b>       <U2023>   TRIANGULAR BULLET
<..>       <U2025>   TWO DOT LEADER
<.3>       <U2026>   HORIZONTAL ELLIPSIS
<.->       <U2027>   HYPHENATION POINT
<linesep>  <U2028>   LINE SEPARATOR
<parsep>   <U2029>   PARAGRAPH SEPARATOR
<%0>       <U2030>   PER MILLE SIGN
<1'>       <U2032>   PRIME
<2'>       <U2033>   DOUBLE PRIME
<3'>       <U2034>   TRIPLE PRIME
<1">       <U2035>   REVERSED PRIME
<2">       <U2036>   REVERSED DOUBLE PRIME
<3">       <U2037>   REVERSED TRIPLE PRIME
<Ca>       <U2038>   CARET
<<1>       <U2039>   SINGLE LEFT-POINTING ANGLE QUOTATION MARK
</>1>      <U203A>   SINGLE RIGHT-POINTING ANGLE QUOTATION MARK
<:X>       <U203B>   REFERENCE MARK
<!*2>      <U203C>   DOUBLE EXCLAMATION MARK
<'->       <U203E>   OVERLINE
<-b>       <U2043>   HYPHEN BULLET
<//f>      <U2044>   FRACTION SLASH
<0S>       <U2070>   SUPERSCRIPT ZERO
<4S>       <U2074>   SUPERSCRIPT FOUR
<5S>       <U2075>   SUPERSCRIPT FIVE
<6S>       <U2076>   SUPERSCRIPT SIX
<7S>       <U2077>   SUPERSCRIPT SEVEN
<8S>       <U2078>   SUPERSCRIPT EIGHT
<9S>       <U2079>   SUPERSCRIPT NINE
<+S>       <U207A>   SUPERSCRIPT PLUS SIGN
<-S>       <U207B>   SUPERSCRIPT MINUS
<=S>       <U207C>   SUPERSCRIPT EQUALS SIGN
<(S>       <U207D>   SUPERSCRIPT LEFT PARENTHESIS
<)S>       <U207E>   SUPERSCRIPT RIGHT PARENTHESIS
<nS>       <U207F>   SUPERSCRIPT LATIN SMALL LETTER N
<0s>       <U2080>   SUBSCRIPT ZERO
<1s>       <U2081>   SUBSCRIPT ONE
<2s>       <U2082>   SUBSCRIPT TWO
<3s>       <U2083>   SUBSCRIPT THREE
<4s>       <U2084>   SUBSCRIPT FOUR
<5s>       <U2085>   SUBSCRIPT FIVE
<6s>       <U2086>   SUBSCRIPT SIX
<7s>       <U2087>   SUBSCRIPT SEVEN
<8s>       <U2088>   SUBSCRIPT EIGHT
<9s>       <U2089>   SUBSCRIPT NINE
<+s>       <U208A>   SUBSCRIPT PLUS SIGN
<-s>       <U208B>   SUBSCRIPT MINUS
<=s>       <U208C>   SUBSCRIPT EQUALS SIGN
<(s>       <U208D>   SUBSCRIPT LEFT PARENTHESIS
<)s>       <U208E>   SUBSCRIPT RIGHT PARENTHESIS
<Ff>       <U20A3>   FRENCH FRANC SIGN
<Li>       <U20A4>   LIRA SIGN
<Pt>       <U20A7>   PESETA SIGN
<W=>       <U20A9>   WON SIGN
<Eu>       <U20AC>   EURO SIGN
<"7>       <U20D1>   COMBINING RIGHT HARPOON ABOVE
<oC>       <U2103>   DEGREE CELSIUS
<co>       <U2105>   CARE OF
<oF>       <U2109>   DEGREE FAHRENHEIT
<N0>       <U2116>   NUMERO SIGN
<PO>       <U2117>   SOUND RECORDING COPYRIGHT
<Rx>       <U211E>   PRESCRIPTION TAKE
<SM>       <U2120>   SERVICE MARK
<TM>       <U2122>   TRADE MARK SIGN
```



```
<Om>       <U2126>   OHM SIGN
<AO>       <U212B>   ANGSTROM SIGN
<Est>      <U212E>   ESTIMATED SYMBOL
<13>       <U2153>   VULGAR FRACTION ONE THIRD
<23>       <U2154>   VULGAR FRACTION TWO THIRDS
<15>       <U2155>   VULGAR FRACTION ONE FIFTH
<25>       <U2156>   VULGAR FRACTION TWO FIFTHS
<35>       <U2157>   VULGAR FRACTION THREE FIFTHS
<45>       <U2158>   VULGAR FRACTION FOUR FIFTHS
<16>       <U2159>   VULGAR FRACTION ONE SIXTH
<56>       <U215A>   VULGAR FRACTION FIVE SIXTHS
<18>       <U215B>   VULGAR FRACTION ONE EIGHTH
<38>       <U215C>   VULGAR FRACTION THREE EIGHTHS
<58>       <U215D>   VULGAR FRACTION FIVE EIGHTHS
<78>       <U215E>   VULGAR FRACTION SEVEN EIGHTHS
<1R>       <U2160>   ROMAN NUMERAL ONE
<2R>       <U2161>   ROMAN NUMERAL TWO
<3R>       <U2162>   ROMAN NUMERAL THREE
<4R>       <U2163>   ROMAN NUMERAL FOUR
<5R>       <U2164>   ROMAN NUMERAL FIVE
<6R>       <U2165>   ROMAN NUMERAL SIX
<7R>       <U2166>   ROMAN NUMERAL SEVEN
<8R>       <U2167>   ROMAN NUMERAL EIGHT
<9R>       <U2168>   ROMAN NUMERAL NINE
<aR>       <U2169>   ROMAN NUMERAL TEN
<bR>       <U216A>   ROMAN NUMERAL ELEVEN
<cR>       <U216B>   ROMAN NUMERAL TWELVE
<50R>      <U216C>   ROMAN NUMERAL FIFTY
<100R>     <U216D>   ROMAN NUMERAL ONE HUNDRED
<500R>     <U216E>   ROMAN NUMERAL FIVE HUNDRED
<1000R>    <U216F>   ROMAN NUMERAL ONE THOUSAND
<1r>       <U2170>   SMALL ROMAN NUMERAL ONE
<2r>       <U2171>   SMALL ROMAN NUMERAL TWO
<3r>       <U2172>   SMALL ROMAN NUMERAL THREE
<4r>       <U2173>   SMALL ROMAN NUMERAL FOUR
<5r>       <U2174>   SMALL ROMAN NUMERAL FIVE
<6r>       <U2175>   SMALL ROMAN NUMERAL SIX
<7r>       <U2176>   SMALL ROMAN NUMERAL SEVEN
<8r>       <U2177>   SMALL ROMAN NUMERAL EIGHT
<9r>       <U2178>   SMALL ROMAN NUMERAL NINE
<ar>       <U2179>   SMALL ROMAN NUMERAL TEN
<br>       <U217A>   SMALL ROMAN NUMERAL ELEVEN
<cr>       <U217B>   SMALL ROMAN NUMERAL TWELVE
<50r>      <U217C>   SMALL ROMAN NUMERAL FIFTY
<100r>     <U217D>   SMALL ROMAN NUMERAL ONE HUNDRED
<500r>     <U217E>   SMALL ROMAN NUMERAL FIVE HUNDRED
<1000r>    <U217F>   SMALL ROMAN NUMERAL ONE THOUSAND
<1000RCD>  <U2180>   ROMAN NUMERAL ONE THOUSAND C D
<5000R>    <U2181>   ROMAN NUMERAL FIVE THOUSAND
<10000R>   <U2182>   ROMAN NUMERAL TEN THOUSAND
<<->       <U2190>   LEFTWARDS ARROW
<-!>       <U2191>   UPWARDS ARROW
<-/>>      <U2192>   RIGHTWARDS ARROW
<-v>       <U2193>   DOWNWARDS ARROW
<</>>      <U2194>   LEFT RIGHT ARROW
<UD>       <U2195>   UP DOWN ARROW
<<!!>      <U2196>   NORTH WEST ARROW
</////>>   <U2197>   NORTH EAST ARROW
<!!/>>     <U2198>   SOUTH EAST ARROW
<<////>    <U2199>   SOUTH WEST ARROW
<UD->      <U21A8>   UP DOWN ARROW WITH BASE
</>V>      <U21C0>   RIGHTWARDS HARPOON WITH BARB UPWARDS
<<=>       <U21D0>   LEFTWARDS DOUBLE ARROW
<=/>>      <U21D2>   RIGHTWARDS DOUBLE ARROW
<==>       <U21D4>   LEFT RIGHT DOUBLE ARROW
<FA>       <U2200>   FOR ALL
<dP>       <U2202>   PARTIAL DIFFERENTIAL
<TE>       <U2203>   THERE EXISTS
<//0>      <U2205>   EMPTY SET
<DE>       <U2206>   INCREMENT
<NB>       <U2207>   NABLA
<(->       <U2208>   ELEMENT OF
<-)>       <U220B>   CONTAINS AS MEMBER
<FP>       <U220E>   END OF PROOF
```


<*P>	<U220F>	N-ARY PRODUCT
<+Z>	<U2211>	N-ARY SUMMATION
<-2>	<U2212>	MINUS SIGN
<-+>	<U2213>	MINUS-OR-PLUS SIGN
<.+>	<U2214>	DOT PLUS
<*->	<U2217>	ASTERISK OPERATOR
<Ob>	<U2218>	RING OPERATOR
<Sb>	<U2219>	BULLET OPERATOR
<RT>	<U221A>	SQUARE ROOT
<0(>	<U221D>	PROPORTIONAL TO
<00>	<U221E>	INFINITY
<-L>	<U221F>	RIGHT ANGLE
<-V>	<U2220>	ANGLE
<PP>	<U2225>	PARALLEL TO
<AN>	<U2227>	LOGICAL AND
<OR>	<U2228>	LOGICAL OR
<(U>	<U2229>	INTERSECTION
<)U>	<U222A>	UNION
<In>	<U222B>	INTEGRAL
<DI>	<U222C>	DOUBLE INTEGRAL
<Io>	<U222E>	CONTOUR INTEGRAL
<.:>	<U2234>	THEREFORE
<:.>	<U2235>	BECAUSE
<:R>	<U2236>	RATIO
<::>	<U2237>	PROPORTION
<?1>	<U223C>	TILDE OPERATOR
<CG>	<U223E>	INVERTED LAZY S
<?->	<U2243>	ASYMPTOTICALLY EQUAL TO
<?=>	<U2245>	APPROXIMATELY EQUAL TO
<?2>	<U2248>	ALMOST EQUAL TO
<=?>	<U224C>	ALL EQUAL TO
<HI>	<U2253>	IMAGE OF OR APPROXIMATELY EQUAL TO
<!=>	<U2260>	NOT EQUAL TO
<=3>	<U2261>	IDENTICAL TO
<=<>	<U2264>	LESS-THAN OR EQUAL TO
</>=>	<U2265>	GREATER-THAN OR EQUAL TO
<<*>	<U226A>	MUCH LESS-THAN
<*/>>	<U226B>	MUCH GREATER-THAN
<!<>	<U226E>	NOT LESS-THAN
<!/>>	<U226F>	NOT GREATER-THAN
<(C>	<U2282>	SUBSET OF
<)C>	<U2283>	SUPERSET OF
<(_>	<U2286>	SUBSET OF OR EQUAL TO
<)_>	<U2287>	SUPERSET OF OR EQUAL TO
<0.>	<U2299>	CIRCLED DOT OPERATOR
<02>	<U229A>	CIRCLED RING OPERATOR
<-T>	<U22A5>	UP TACK
<.P>	<U22C5>	DOT OPERATOR
<:3>	<U22EE>	VERTICAL ELLIPSIS
<Eh>	<U2302>	HOUSE
<<7>	<U2308>	LEFT CEILING
</>7>	<U2309>	RIGHT CEILING
<7<>	<U230A>	LEFT FLOOR
<7/>>	<U230B>	RIGHT FLOOR
<NI>	<U2310>	REVERSED NOT SIGN
<(A>	<U2312>	ARC
<TR>	<U2315>	TELEPHONE RECORDER
<88>	<U2318>	PLACE OF INTEREST SIGN
<Iu>	<U2320>	TOP HALF INTEGRAL
<Il>	<U2321>	BOTTOM HALF INTEGRAL
<<//>	<U2329>	LEFT-POINTING ANGLE BRACKET
<///>>	<U232A>	RIGHT-POINTING ANGLE BRACKET
<Vs>	<U2423>	OPEN BOX
<1h>	<U2440>	OCR HOOK
<3h>	<U2441>	OCR CHAIR
<2h>	<U2442>	OCR FORK
<4h>	<U2443>	OCR INVERTED FORK
<1j>	<U2446>	OCR BRANCH BANK IDENTIFICATION
<2j>	<U2447>	OCR AMOUNT OF CHECK
<3j>	<U2448>	OCR DASH
<4j>	<U2449>	OCR CUSTOMER ACCOUNT NUMBER
<1-o>	<U2460>	CIRCLED DIGIT ONE
<2-o>	<U2461>	CIRCLED DIGIT TWO
<3-o>	<U2462>	CIRCLED DIGIT THREE


```
<4-o>      <U2463>   CIRCLED DIGIT FOUR
<5-o>      <U2464>   CIRCLED DIGIT FIVE
<6-o>      <U2465>   CIRCLED DIGIT SIX
<7-o>      <U2466>   CIRCLED DIGIT SEVEN
<8-o>      <U2467>   CIRCLED DIGIT EIGHT
<9-o>      <U2468>   CIRCLED DIGIT NINE
<10-o>     <U2469>   CIRCLED NUMBER TEN
<11-o>     <U246A>   CIRCLED NUMBER ELEVEN
<12-o>     <U246B>   CIRCLED NUMBER TWELVE
<13-o>     <U246C>   CIRCLED NUMBER THIRTEEN
<14-o>     <U246D>   CIRCLED NUMBER FOURTEEN
<15-o>     <U246E>   CIRCLED NUMBER FIFTEEN
<16-o>     <U246F>   CIRCLED NUMBER SIXTEEN
<17-o>     <U2470>   CIRCLED NUMBER SEVENTEEN
<18-o>     <U2471>   CIRCLED NUMBER EIGHTEEN
<19-o>     <U2472>   CIRCLED NUMBER NINETEEN
<20-o>     <U2473>   CIRCLED NUMBER TWENTY
<(1)>      <U2474>   PARENTHESIZED DIGIT ONE
<(2)>      <U2475>   PARENTHESIZED DIGIT TWO
<(3)>      <U2476>   PARENTHESIZED DIGIT THREE
<(4)>      <U2477>   PARENTHESIZED DIGIT FOUR
<(5)>      <U2478>   PARENTHESIZED DIGIT FIVE
<(6)>      <U2479>   PARENTHESIZED DIGIT SIX
<(7)>      <U247A>   PARENTHESIZED DIGIT SEVEN
<(8)>      <U247B>   PARENTHESIZED DIGIT EIGHT
<(9)>      <U247C>   PARENTHESIZED DIGIT NINE
<(10)>     <U247D>   PARENTHESIZED NUMBER TEN
<(11)>     <U247E>   PARENTHESIZED NUMBER ELEVEN
<(12)>     <U247F>   PARENTHESIZED NUMBER TWELVE
<(13)>     <U2480>   PARENTHESIZED NUMBER THIRTEEN
<(14)>     <U2481>   PARENTHESIZED NUMBER FOURTEEN
<(15)>     <U2482>   PARENTHESIZED NUMBER FIFTEEN
<(16)>     <U2483>   PARENTHESIZED NUMBER SIXTEEN
<(17)>     <U2484>   PARENTHESIZED NUMBER SEVENTEEN
<(18)>     <U2485>   PARENTHESIZED NUMBER EIGHTEEN
<(19)>     <U2486>   PARENTHESIZED NUMBER NINETEEN
<(20)>     <U2487>   PARENTHESIZED NUMBER TWENTY
<1.>       <U2488>   DIGIT ONE FULL STOP
<2.>       <U2489>   DIGIT TWO FULL STOP
<3.>       <U248A>   DIGIT THREE FULL STOP
<4.>       <U248B>   DIGIT FOUR FULL STOP
<5.>       <U248C>   DIGIT FIVE FULL STOP
<6.>       <U248D>   DIGIT SIX FULL STOP
<7.>       <U248E>   DIGIT SEVEN FULL STOP
<8.>       <U248F>   DIGIT EIGHT FULL STOP
<9.>       <U2490>   DIGIT NINE FULL STOP
<10.>      <U2491>   NUMBER TEN FULL STOP
<11.>      <U2492>   NUMBER ELEVEN FULL STOP
<12.>      <U2493>   NUMBER TWELVE FULL STOP
<13.>      <U2494>   NUMBER THIRTEEN FULL STOP
<14.>      <U2495>   NUMBER FOURTEEN FULL STOP
<15.>      <U2496>   NUMBER FIFTEEN FULL STOP
<16.>      <U2497>   NUMBER SIXTEEN FULL STOP
<17.>      <U2498>   NUMBER SEVENTEEN FULL STOP
<18.>      <U2499>   NUMBER EIGHTEEN FULL STOP
<19.>      <U249A>   NUMBER NINETEEN FULL STOP
<20.>      <U249B>   NUMBER TWENTY FULL STOP
<(a)>      <U249C>   PARENTHESIZED LATIN SMALL LETTER A
<(b)>      <U249D>   PARENTHESIZED LATIN SMALL LETTER B
<(c)>      <U249E>   PARENTHESIZED LATIN SMALL LETTER C
<(d)>      <U249F>   PARENTHESIZED LATIN SMALL LETTER D
<(e)>      <U24A0>   PARENTHESIZED LATIN SMALL LETTER E
<(f)>      <U24A1>   PARENTHESIZED LATIN SMALL LETTER F
<(g)>      <U24A2>   PARENTHESIZED LATIN SMALL LETTER G
<(h)>      <U24A3>   PARENTHESIZED LATIN SMALL LETTER H
<(i)>      <U24A4>   PARENTHESIZED LATIN SMALL LETTER I
<(j)>      <U24A5>   PARENTHESIZED LATIN SMALL LETTER J
<(k)>      <U24A6>   PARENTHESIZED LATIN SMALL LETTER K
<(l)>      <U24A7>   PARENTHESIZED LATIN SMALL LETTER L
<(m)>      <U24A8>   PARENTHESIZED LATIN SMALL LETTER M
<(n)>      <U24A9>   PARENTHESIZED LATIN SMALL LETTER N
<(o)>      <U24AA>   PARENTHESIZED LATIN SMALL LETTER O
<(p)>      <U24AB>   PARENTHESIZED LATIN SMALL LETTER P
<(q)>      <U24AC>   PARENTHESIZED LATIN SMALL LETTER Q
```



```
<(r)>      <U24AD>   PARENTHESIZED LATIN SMALL LETTER R
<(s)>      <U24AE>   PARENTHESIZED LATIN SMALL LETTER S
<(t)>      <U24AF>   PARENTHESIZED LATIN SMALL LETTER T
<(u)>      <U24B0>   PARENTHESIZED LATIN SMALL LETTER U
<(v)>      <U24B1>   PARENTHESIZED LATIN SMALL LETTER V
<(w)>      <U24B2>   PARENTHESIZED LATIN SMALL LETTER W
<(x)>      <U24B3>   PARENTHESIZED LATIN SMALL LETTER X
<(y)>      <U24B4>   PARENTHESIZED LATIN SMALL LETTER Y
<(z)>      <U24B5>   PARENTHESIZED LATIN SMALL LETTER Z
<A-o>      <U24B6>   CIRCLED LATIN CAPITAL LETTER A
<B-o>      <U24B7>   CIRCLED LATIN CAPITAL LETTER B
<C-o>      <U24B8>   CIRCLED LATIN CAPITAL LETTER C
<D-o>      <U24B9>   CIRCLED LATIN CAPITAL LETTER D
<E-o>      <U24BA>   CIRCLED LATIN CAPITAL LETTER E
<F-o>      <U24BB>   CIRCLED LATIN CAPITAL LETTER F
<G-o>      <U24BC>   CIRCLED LATIN CAPITAL LETTER G
<H-o>      <U24BD>   CIRCLED LATIN CAPITAL LETTER H
<I-o>      <U24BE>   CIRCLED LATIN CAPITAL LETTER I
<J-o>      <U24BF>   CIRCLED LATIN CAPITAL LETTER J
<K-o>      <U24C0>   CIRCLED LATIN CAPITAL LETTER K
<L-o>      <U24C1>   CIRCLED LATIN CAPITAL LETTER L
<M-o>      <U24C2>   CIRCLED LATIN CAPITAL LETTER M
<N-o>      <U24C3>   CIRCLED LATIN CAPITAL LETTER N
<O-o>      <U24C4>   CIRCLED LATIN CAPITAL LETTER O
<P-o>      <U24C5>   CIRCLED LATIN CAPITAL LETTER P
<Q-o>      <U24C6>   CIRCLED LATIN CAPITAL LETTER Q
<R-o>      <U24C7>   CIRCLED LATIN CAPITAL LETTER R
<S-o>      <U24C8>   CIRCLED LATIN CAPITAL LETTER S
<T-o>      <U24C9>   CIRCLED LATIN CAPITAL LETTER T
<U-o>      <U24CA>   CIRCLED LATIN CAPITAL LETTER U
<V-o>      <U24CB>   CIRCLED LATIN CAPITAL LETTER V
<W-o>      <U24CC>   CIRCLED LATIN CAPITAL LETTER W
<X-o>      <U24CD>   CIRCLED LATIN CAPITAL LETTER X
<Y-o>      <U24CE>   CIRCLED LATIN CAPITAL LETTER Y
<Z-o>      <U24CF>   CIRCLED LATIN CAPITAL LETTER Z
<a-o>      <U24D0>   CIRCLED LATIN SMALL LETTER A
<b-o>      <U24D1>   CIRCLED LATIN SMALL LETTER B
<c-o>      <U24D2>   CIRCLED LATIN SMALL LETTER C
<d-o>      <U24D3>   CIRCLED LATIN SMALL LETTER D
<e-o>      <U24D4>   CIRCLED LATIN SMALL LETTER E
<f-o>      <U24D5>   CIRCLED LATIN SMALL LETTER F
<g-o>      <U24D6>   CIRCLED LATIN SMALL LETTER G
<h-o>      <U24D7>   CIRCLED LATIN SMALL LETTER H
<i-o>      <U24D8>   CIRCLED LATIN SMALL LETTER I
<j-o>      <U24D9>   CIRCLED LATIN SMALL LETTER J
<k-o>      <U24DA>   CIRCLED LATIN SMALL LETTER K
<l-o>      <U24DB>   CIRCLED LATIN SMALL LETTER L
<m-o>      <U24DC>   CIRCLED LATIN SMALL LETTER M
<n-o>      <U24DD>   CIRCLED LATIN SMALL LETTER N
<o-o>      <U24DE>   CIRCLED LATIN SMALL LETTER O
<p-o>      <U24DF>   CIRCLED LATIN SMALL LETTER P
<q-o>      <U24E0>   CIRCLED LATIN SMALL LETTER Q
<r-o>      <U24E1>   CIRCLED LATIN SMALL LETTER R
<s-o>      <U24E2>   CIRCLED LATIN SMALL LETTER S
<t-o>      <U24E3>   CIRCLED LATIN SMALL LETTER T
<u-o>      <U24E4>   CIRCLED LATIN SMALL LETTER U
<v-o>      <U24E5>   CIRCLED LATIN SMALL LETTER V
<w-o>      <U24E6>   CIRCLED LATIN SMALL LETTER W
<x-o>      <U24E7>   CIRCLED LATIN SMALL LETTER X
<y-o>      <U24E8>   CIRCLED LATIN SMALL LETTER Y
<z-o>      <U24E9>   CIRCLED LATIN SMALL LETTER Z
<0-o>      <U24EA>   CIRCLED DIGIT ZERO
<hh>       <U2500>   BOX DRAWINGS LIGHT HORIZONTAL
<HH->      <U2501>   BOX DRAWINGS HEAVY HORIZONTAL
<vv>       <U2502>   BOX DRAWINGS LIGHT VERTICAL
<VV->      <U2503>   BOX DRAWINGS HEAVY VERTICAL
<3->       <U2504>   BOX DRAWINGS LIGHT TRIPLE DASH HORIZONTAL
<3_>       <U2505>   BOX DRAWINGS HEAVY TRIPLE DASH HORIZONTAL
<3!>       <U2506>   BOX DRAWINGS LIGHT TRIPLE DASH VERTICAL
<3//>      <U2507>   BOX DRAWINGS HEAVY TRIPLE DASH VERTICAL
<4->       <U2508>   BOX DRAWINGS LIGHT QUADRUPLE DASH HORIZONTAL
<4_>       <U2509>   BOX DRAWINGS HEAVY QUADRUPLE DASH HORIZONTAL
<4!>       <U250A>   BOX DRAWINGS LIGHT QUADRUPLE DASH VERTICAL
<4//>      <U250B>   BOX DRAWINGS HEAVY QUADRUPLE DASH VERTICAL
```



```
<dr>       <U250C>   BOX DRAWINGS LIGHT DOWN AND RIGHT
<dR->      <U250D>   BOX DRAWINGS DOWN LIGHT AND RIGHT HEAVY
<Dr->      <U250E>   BOX DRAWINGS DOWN HEAVY AND RIGHT LIGHT
<DR->      <U250F>   BOX DRAWINGS HEAVY DOWN AND RIGHT
<dl>       <U2510>   BOX DRAWINGS LIGHT DOWN AND LEFT
<dL->      <U2511>   BOX DRAWINGS DOWN LIGHT AND LEFT HEAVY
<Dl->      <U2512>   BOX DRAWINGS DOWN HEAVY AND LEFT LIGHT
<LD->      <U2513>   BOX DRAWINGS HEAVY DOWN AND LEFT
<ur>       <U2514>   BOX DRAWINGS LIGHT UP AND RIGHT
<uR->      <U2515>   BOX DRAWINGS UP LIGHT AND RIGHT HEAVY
<Ur->      <U2516>   BOX DRAWINGS UP HEAVY AND RIGHT LIGHT
<UR->      <U2517>   BOX DRAWINGS HEAVY UP AND RIGHT
<ul>       <U2518>   BOX DRAWINGS LIGHT UP AND LEFT
<uL->      <U2519>   BOX DRAWINGS UP LIGHT AND LEFT HEAVY
<Ul->      <U251A>   BOX DRAWINGS UP HEAVY AND LEFT LIGHT
<UL->      <U251B>   BOX DRAWINGS HEAVY UP AND LEFT
<vr>       <U251C>   BOX DRAWINGS LIGHT VERTICAL AND RIGHT
<vR->      <U251D>   BOX DRAWINGS VERTICAL LIGHT AND RIGHT HEAVY
<Udr>      <U251E>   BOX DRAWINGS UP HEAVY AND RIGHT DOWN LIGHT
<uDr>      <U251F>   BOX DRAWINGS DOWN HEAVY AND RIGHT UP LIGHT
<Vr->      <U2520>   BOX DRAWINGS VERTICAL HEAVY AND RIGHT LIGHT
<UdR>      <U2521>   BOX DRAWINGS DOWN LIGHT AND RIGHT UP HEAVY
<uDR>      <U2522>   BOX DRAWINGS UP LIGHT AND RIGHT DOWN HEAVY
<VR->      <U2523>   BOX DRAWINGS HEAVY VERTICAL AND RIGHT
<vl>       <U2524>   BOX DRAWINGS LIGHT VERTICAL AND LEFT
<vL->      <U2525>   BOX DRAWINGS VERTICAL LIGHT AND LEFT HEAVY
<Udl>      <U2526>   BOX DRAWINGS UP HEAVY AND LEFT DOWN LIGHT
<uDl>      <U2527>   BOX DRAWINGS DOWN HEAVY AND LEFT UP LIGHT
<Vl->      <U2528>   BOX DRAWINGS VERTICAL HEAVY AND LEFT LIGHT
<UdL>      <U2529>   BOX DRAWINGS DOWN LIGHT AND LEFT UP HEAVY
<uDL>      <U252A>   BOX DRAWINGS UP LIGHT AND LEFT DOWN HEAVY
<VL->      <U252B>   BOX DRAWINGS HEAVY VERTICAL AND LEFT
<dh>       <U252C>   BOX DRAWINGS LIGHT DOWN AND HORIZONTAL
<dLr>      <U252D>   BOX DRAWINGS LEFT HEAVY AND RIGHT DOWN LIGHT
<dlR>      <U252E>   BOX DRAWINGS RIGHT HEAVY AND LEFT DOWN LIGHT
<dH->      <U252F>   BOX DRAWINGS DOWN LIGHT AND HORIZONTAL HEAVY
<Dh->      <U2530>   BOX DRAWINGS DOWN HEAVY AND HORIZONTAL LIGHT
<DLr>      <U2531>   BOX DRAWINGS RIGHT LIGHT AND LEFT DOWN HEAVY
<DlR>      <U2532>   BOX DRAWINGS LEFT LIGHT AND RIGHT DOWN HEAVY
<DH->      <U2533>   BOX DRAWINGS HEAVY DOWN AND HORIZONTAL
<uh>       <U2534>   BOX DRAWINGS LIGHT UP AND HORIZONTAL
<uLr>      <U2535>   BOX DRAWINGS LEFT HEAVY AND RIGHT UP LIGHT
<ulR>      <U2536>   BOX DRAWINGS RIGHT HEAVY AND LEFT UP LIGHT
<uH->      <U2537>   BOX DRAWINGS UP LIGHT AND HORIZONTAL HEAVY
<Uh->      <U2538>   BOX DRAWINGS UP HEAVY AND HORIZONTAL LIGHT
<ULr>      <U2539>   BOX DRAWINGS RIGHT LIGHT AND LEFT UP HEAVY
<UlR>      <U253A>   BOX DRAWINGS LEFT LIGHT AND RIGHT UP HEAVY
<UH->      <U253B>   BOX DRAWINGS HEAVY UP AND HORIZONTAL
<vh>       <U253C>   BOX DRAWINGS LIGHT VERTICAL AND HORIZONTAL
<vLr>      <U253D>   BOX DRAWINGS LEFT HEAVY AND RIGHT VERTICAL LIGHT
<vlR>      <U253E>   BOX DRAWINGS RIGHT HEAVY AND LEFT VERTICAL LIGHT
<vH->      <U253F>   BOX DRAWINGS VERTICAL LIGHT AND HORIZONTAL HEAVY
<Udh>      <U2540>   BOX DRAWINGS UP HEAVY AND DOWN HORIZONTAL LIGHT
<uDh>      <U2541>   BOX DRAWINGS DOWN HEAVY AND UP HORIZONTAL LIGHT
<Vh->      <U2542>   BOX DRAWINGS VERTICAL HEAVY AND HORIZONTAL LIGHT
<UdLr>     <U2543>   BOX DRAWINGS LEFT UP HEAVY AND RIGHT DOWN LIGHT
<UdlR>     <U2544>   BOX DRAWINGS RIGHT UP HEAVY AND LEFT DOWN LIGHT
<uDLr>     <U2545>   BOX DRAWINGS LEFT DOWN HEAVY AND RIGHT UP LIGHT
<uDlR>     <U2546>   BOX DRAWINGS RIGHT DOWN HEAVY AND LEFT UP LIGHT
<UdH>      <U2547>   BOX DRAWINGS DOWN LIGHT AND UP HORIZONTAL HEAVY
<uDH>      <U2548>   BOX DRAWINGS UP LIGHT AND DOWN HORIZONTAL HEAVY
<VLr>      <U2549>   BOX DRAWINGS RIGHT LIGHT AND LEFT VERTICAL HEAVY
<VlR>      <U254A>   BOX DRAWINGS LEFT LIGHT AND RIGHT VERTICAL HEAVY
<VH->      <U254B>   BOX DRAWINGS HEAVY VERTICAL AND HORIZONTAL
<HH>       <U2550>   BOX DRAWINGS DOUBLE HORIZONTAL
<VV>       <U2551>   BOX DRAWINGS DOUBLE VERTICAL
<dR>       <U2552>   BOX DRAWINGS DOWN SINGLE AND RIGHT DOUBLE
<Dr>       <U2553>   BOX DRAWINGS DOWN DOUBLE AND RIGHT SINGLE
<DR>       <U2554>   BOX DRAWINGS DOUBLE DOWN AND RIGHT
<dL>       <U2555>   BOX DRAWINGS DOWN SINGLE AND LEFT DOUBLE
<Dl>       <U2556>   BOX DRAWINGS DOWN DOUBLE AND LEFT SINGLE
<LD>       <U2557>   BOX DRAWINGS DOUBLE DOWN AND LEFT
<uR>       <U2558>   BOX DRAWINGS UP SINGLE AND RIGHT DOUBLE
<Ur>       <U2559>   BOX DRAWINGS UP DOUBLE AND RIGHT SINGLE
```

```
<UR>       <U255A>   BOX DRAWINGS DOUBLE UP AND RIGHT
<uL>       <U255B>   BOX DRAWINGS UP SINGLE AND LEFT DOUBLE
<Ul>       <U255C>   BOX DRAWINGS UP DOUBLE AND LEFT SINGLE
<UL>       <U255D>   BOX DRAWINGS DOUBLE UP AND LEFT
<vR>       <U255E>   BOX DRAWINGS VERTICAL SINGLE AND RIGHT DOUBLE
<Vr>       <U255F>   BOX DRAWINGS VERTICAL DOUBLE AND RIGHT SINGLE
<VR>       <U2560>   BOX DRAWINGS DOUBLE VERTICAL AND RIGHT
<vL>       <U2561>   BOX DRAWINGS VERTICAL SINGLE AND LEFT DOUBLE
<Vl>       <U2562>   BOX DRAWINGS VERTICAL DOUBLE AND LEFT SINGLE
<VL>       <U2563>   BOX DRAWINGS DOUBLE VERTICAL AND LEFT
<dH>       <U2564>   BOX DRAWINGS DOWN SINGLE AND HORIZONTAL DOUBLE
<Dh>       <U2565>   BOX DRAWINGS DOWN DOUBLE AND HORIZONTAL SINGLE
<DH>       <U2566>   BOX DRAWINGS DOUBLE DOWN AND HORIZONTAL
<uH>       <U2567>   BOX DRAWINGS UP SINGLE AND HORIZONTAL DOUBLE
<Uh>       <U2568>   BOX DRAWINGS UP DOUBLE AND HORIZONTAL SINGLE
<UH>       <U2569>   BOX DRAWINGS DOUBLE UP AND HORIZONTAL
<vH>       <U256A>   BOX DRAWINGS VERTICAL SINGLE AND HORIZONTAL DOUBLE
<Vh>       <U256B>   BOX DRAWINGS VERTICAL DOUBLE AND HORIZONTAL SINGLE
<VH>       <U256C>   BOX DRAWINGS DOUBLE VERTICAL AND HORIZONTAL
<FD>       <U2571>   BOX DRAWINGS LIGHT DIAGONAL UPPER RIGHT TO LOWER LEFT
<BD>       <U2572>   BOX DRAWINGS LIGHT DIAGONAL UPPER LEFT TO LOWER RIGHT
<TB>       <U2580>   UPPER HALF BLOCK
<LB>       <U2584>   LOWER HALF BLOCK
<FB>       <U2588>   FULL BLOCK
<lB>       <U258C>   LEFT HALF BLOCK
<RB>       <U2590>   RIGHT HALF BLOCK
<.S>       <U2591>   LIGHT SHADE
<:S>       <U2592>   MEDIUM SHADE
<?S>       <U2593>   DARK SHADE
<fS>       <U25A0>   BLACK SQUARE
<OS>       <U25A1>   WHITE SQUARE
<RO>       <U25A2>   WHITE SQUARE WITH ROUNDED CORNERS
<Rr>       <U25A3>   WHITE SQUARE CONTAINING BLACK SMALL SQUARE
<RF>       <U25A4>   SQUARE WITH HORIZONTAL FILL
<RY>       <U25A5>   SQUARE WITH VERTICAL FILL
<RH>       <U25A6>   SQUARE WITH ORTHOGONAL CROSSHATCH FILL
<RZ>       <U25A7>   SQUARE WITH UPPER LEFT TO LOWER RIGHT FILL
<RK>       <U25A8>   SQUARE WITH UPPER RIGHT TO LOWER LEFT FILL
<RX>       <U25A9>   SQUARE WITH DIAGONAL CROSSHATCH FILL
<sB>       <U25AA>   BLACK SMALL SQUARE
<SR>       <U25AC>   BLACK RECTANGLE
<Or>       <U25AD>   WHITE RECTANGLE
<UT>       <U25B2>   BLACK UP-POINTING TRIANGLE
<uT>       <U25B3>   WHITE UP-POINTING TRIANGLE
<Tr>       <U25B7>   WHITE RIGHT-POINTING TRIANGLE
<PR>       <U25BA>   BLACK RIGHT-POINTING POINTER
<Dt>       <U25BC>   BLACK DOWN-POINTING TRIANGLE
<dT>       <U25BD>   WHITE DOWN-POINTING TRIANGLE
<Tl>       <U25C1>   WHITE LEFT-POINTING TRIANGLE
<PL>       <U25C4>   BLACK LEFT-POINTING POINTER
<Db>       <U25C6>   BLACK DIAMOND
<Dw>       <U25C7>   WHITE DIAMOND
<LZ>       <U25CA>   LOZENGE
<0m>       <U25CB>   WHITE CIRCLE
<0o>       <U25CE>   BULLSEYE
<0M>       <U25CF>   BLACK CIRCLE
<0L>       <U25D0>   CIRCLE WITH LEFT HALF BLACK
<0R>       <U25D1>   CIRCLE WITH RIGHT HALF BLACK
<Sn>       <U25D8>   INVERSE BULLET
<Ic>       <U25D9>   INVERSE WHITE CIRCLE
<Fd>       <U25E2>   BLACK LOWER RIGHT TRIANGLE
<Bd>       <U25E3>   BLACK LOWER LEFT TRIANGLE
<Ci>       <U25EF>   LARGE CIRCLE
<*2>       <U2605>   BLACK STAR
<*1>       <U2606>   WHITE STAR
<TEL>      <U260E>   BLACK TELEPHONE
<tel>      <U260F>   WHITE TELEPHONE
<<H>       <U261C>   WHITE LEFT POINTING INDEX
</>H>      <U261E>   WHITE RIGHT POINTING INDEX
<0u>       <U263A>   WHITE SMILING FACE
<0U>       <U263B>   BLACK SMILING FACE
<SU>       <U263C>   WHITE SUN WITH RAYS
<Fm>       <U2640>   FEMALE SIGN
<Ml>       <U2642>   MALE SIGN
```



```
<cS>       <U2660>   BLACK SPADE SUIT
<cH>       <U2661>   WHITE HEART SUIT
<cD>       <U2662>   WHITE DIAMOND SUIT
<cC>       <U2663>   BLACK CLUB SUIT
<cS->      <U2664>   WHITE SPADE SUIT
<cH->      <U2665>   BLACK HEART SUIT
<cD->      <U2666>   BLACK DIAMOND SUIT
<cC->      <U2667>   WHITE CLUB SUIT
<Md>       <U2669>   QUARTER NOTE
<M8>       <U266A>   EIGHTH NOTE
<M2>       <U266B>   BEAMED EIGHTH NOTES
<M16>      <U266C>   BEAMED SIXTEENTH NOTES
<Mb>       <U266D>   MUSIC FLAT SIGN
<Mx>       <U266E>   MUSIC NATURAL SIGN
<MX>       <U266F>   MUSIC SHARP SIGN
<OK>       <U2713>   CHECK MARK
<XX>       <U2717>   BALLOT X
<-X>       <U2720>   MALTESE CROSS
<IS>       <U3000>   IDEOGRAPHIC SPACE
<,_>       <U3001>   IDEOGRAPHIC COMMA
<._>       <U3002>   IDEOGRAPHIC FULL STOP
<+">       <U3003>   DITTO MARK
<JIS>      <U3004>   JAPANESE INDUSTRIAL STANDARD SYMBOL
<*_>       <U3005>   IDEOGRAPHIC ITERATION MARK
<;_>       <U3006>   IDEOGRAPHIC CLOSING MARK
<0_>       <U3007>   IDEOGRAPHIC NUMBER ZERO
<<+>       <U300A>   LEFT DOUBLE ANGLE BRACKET
</>+>      <U300B>   RIGHT DOUBLE ANGLE BRACKET
<<'>       <U300C>   LEFT CORNER BRACKET
</>'>      <U300D>   RIGHT CORNER BRACKET
<<">       <U300E>   LEFT WHITE CORNER BRACKET
</>">      <U300F>   RIGHT WHITE CORNER BRACKET
<(">       <U3010>   LEFT BLACK LENTICULAR BRACKET
<)">       <U3011>   RIGHT BLACK LENTICULAR BRACKET
<=T>       <U3012>   POSTAL MARK
<=_>       <U3013>   GETA MARK
<('>       <U3014>   LEFT TORTOISE SHELL BRACKET
<)'>       <U3015>   RIGHT TORTOISE SHELL BRACKET
<(I>       <U3016>   LEFT WHITE LENTICULAR BRACKET
<)I>       <U3017>   RIGHT WHITE LENTICULAR BRACKET
<-?>       <U301C>   WAVE DASH
<=T:)>     <U3020>   POSTAL MARK FACE
<A5>       <U3041>   HIRAGANA LETTER SMALL A
<a5>       <U3042>   HIRAGANA LETTER A
<I5>       <U3043>   HIRAGANA LETTER SMALL I
<i5>       <U3044>   HIRAGANA LETTER I
<U5>       <U3045>   HIRAGANA LETTER SMALL U
<u5>       <U3046>   HIRAGANA LETTER U
<E5>       <U3047>   HIRAGANA LETTER SMALL E
<e5>       <U3048>   HIRAGANA LETTER E
<O5>       <U3049>   HIRAGANA LETTER SMALL O
<o5>       <U304A>   HIRAGANA LETTER O
<ka>       <U304B>   HIRAGANA LETTER KA
<ga>       <U304C>   HIRAGANA LETTER GA
<ki>       <U304D>   HIRAGANA LETTER KI
<gi>       <U304E>   HIRAGANA LETTER GI
<ku>       <U304F>   HIRAGANA LETTER KU
<gu>       <U3050>   HIRAGANA LETTER GU
<ke>       <U3051>   HIRAGANA LETTER KE
<ge>       <U3052>   HIRAGANA LETTER GE
<ko>       <U3053>   HIRAGANA LETTER KO
<go>       <U3054>   HIRAGANA LETTER GO
<sa>       <U3055>   HIRAGANA LETTER SA
<za>       <U3056>   HIRAGANA LETTER ZA
<si>       <U3057>   HIRAGANA LETTER SI
<zi>       <U3058>   HIRAGANA LETTER ZI
<su>       <U3059>   HIRAGANA LETTER SU
<zu>       <U305A>   HIRAGANA LETTER ZU
<se>       <U305B>   HIRAGANA LETTER SE
<ze>       <U305C>   HIRAGANA LETTER ZE
<so>       <U305D>   HIRAGANA LETTER SO
<zo>       <U305E>   HIRAGANA LETTER ZO
<ta>       <U305F>   HIRAGANA LETTER TA
<da>       <U3060>   HIRAGANA LETTER DA
```



```
<ti>       <U3061>   HIRAGANA LETTER TI
<di>       <U3062>   HIRAGANA LETTER DI
<tU>       <U3063>   HIRAGANA LETTER SMALL TU
<tu>       <U3064>   HIRAGANA LETTER TU
<du>       <U3065>   HIRAGANA LETTER DU
<te>       <U3066>   HIRAGANA LETTER TE
<de>       <U3067>   HIRAGANA LETTER DE
<to>       <U3068>   HIRAGANA LETTER TO
<do>       <U3069>   HIRAGANA LETTER DO
<na>       <U306A>   HIRAGANA LETTER NA
<ni>       <U306B>   HIRAGANA LETTER NI
<nu>       <U306C>   HIRAGANA LETTER NU
<ne>       <U306D>   HIRAGANA LETTER NE
<no>       <U306E>   HIRAGANA LETTER NO
<ha>       <U306F>   HIRAGANA LETTER HA
<ba>       <U3070>   HIRAGANA LETTER BA
<pa>       <U3071>   HIRAGANA LETTER PA
<hi>       <U3072>   HIRAGANA LETTER HI
<bi>       <U3073>   HIRAGANA LETTER BI
<pi>       <U3074>   HIRAGANA LETTER PI
<hu>       <U3075>   HIRAGANA LETTER HU
<bu>       <U3076>   HIRAGANA LETTER BU
<pu>       <U3077>   HIRAGANA LETTER PU
<he>       <U3078>   HIRAGANA LETTER HE
<be>       <U3079>   HIRAGANA LETTER BE
<pe>       <U307A>   HIRAGANA LETTER PE
<ho>       <U307B>   HIRAGANA LETTER HO
<bo>       <U307C>   HIRAGANA LETTER BO
<po>       <U307D>   HIRAGANA LETTER PO
<ma>       <U307E>   HIRAGANA LETTER MA
<mi>       <U307F>   HIRAGANA LETTER MI
<mu>       <U3080>   HIRAGANA LETTER MU
<me>       <U3081>   HIRAGANA LETTER ME
<mo>       <U3082>   HIRAGANA LETTER MO
<yA>       <U3083>   HIRAGANA LETTER SMALL YA
<ya>       <U3084>   HIRAGANA LETTER YA
<yU>       <U3085>   HIRAGANA LETTER SMALL YU
<yu>       <U3086>   HIRAGANA LETTER YU
<yO>       <U3087>   HIRAGANA LETTER SMALL YO
<yo>       <U3088>   HIRAGANA LETTER YO
<ra>       <U3089>   HIRAGANA LETTER RA
<ri>       <U308A>   HIRAGANA LETTER RI
<ru>       <U308B>   HIRAGANA LETTER RU
<re>       <U308C>   HIRAGANA LETTER RE
<ro>       <U308D>   HIRAGANA LETTER RO
<wA>       <U308E>   HIRAGANA LETTER SMALL WA
<wa>       <U308F>   HIRAGANA LETTER WA
<wi>       <U3090>   HIRAGANA LETTER WI
<we>       <U3091>   HIRAGANA LETTER WE
<wo>       <U3092>   HIRAGANA LETTER WO
<n5>       <U3093>   HIRAGANA LETTER N
<vu>       <U3094>   HIRAGANA LETTER VU
<"5>       <U309B>   KATAKANA-HIRAGANA VOICED SOUND MARK
<05>       <U309C>   KATAKANA-HIRAGANA SEMI-VOICED SOUND MARK
<*5>       <U309D>   HIRAGANA ITERATION MARK
<+5>       <U309E>   HIRAGANA VOICED ITERATION MARK
<a6>       <U30A1>   KATAKANA LETTER SMALL A
<A6>       <U30A2>   KATAKANA LETTER A
<i6>       <U30A3>   KATAKANA LETTER SMALL I
<I6>       <U30A4>   KATAKANA LETTER I
<u6>       <U30A5>   KATAKANA LETTER SMALL U
<U6>       <U30A6>   KATAKANA LETTER U
<e6>       <U30A7>   KATAKANA LETTER SMALL E
<E6>       <U30A8>   KATAKANA LETTER E
<o6>       <U30A9>   KATAKANA LETTER SMALL O
<O6>       <U30AA>   KATAKANA LETTER O
<Ka>       <U30AB>   KATAKANA LETTER KA
<Ga>       <U30AC>   KATAKANA LETTER GA
<Ki>       <U30AD>   KATAKANA LETTER KI
<Gi>       <U30AE>   KATAKANA LETTER GI
<Ku>       <U30AF>   KATAKANA LETTER KU
<Gu>       <U30B0>   KATAKANA LETTER GU
<Ke>       <U30B1>   KATAKANA LETTER KE
<Ge>       <U30B2>   KATAKANA LETTER GE
```



```
<Ko>       <U30B3>   KATAKANA LETTER KO
<Go>       <U30B4>   KATAKANA LETTER GO
<Sa>       <U30B5>   KATAKANA LETTER SA
<Za>       <U30B6>   KATAKANA LETTER ZA
<Si>       <U30B7>   KATAKANA LETTER SI
<Zi>       <U30B8>   KATAKANA LETTER ZI
<Su>       <U30B9>   KATAKANA LETTER SU
<Zu>       <U30BA>   KATAKANA LETTER ZU
<Se>       <U30BB>   KATAKANA LETTER SE
<Ze>       <U30BC>   KATAKANA LETTER ZE
<So>       <U30BD>   KATAKANA LETTER SO
<Zo>       <U30BE>   KATAKANA LETTER ZO
<Ta>       <U30BF>   KATAKANA LETTER TA
<Da>       <U30C0>   KATAKANA LETTER DA
<Ti>       <U30C1>   KATAKANA LETTER TI
<Di>       <U30C2>   KATAKANA LETTER DI
<TU>       <U30C3>   KATAKANA LETTER SMALL TU
<Tu>       <U30C4>   KATAKANA LETTER TU
<Du>       <U30C5>   KATAKANA LETTER DU
<Te>       <U30C6>   KATAKANA LETTER TE
<De>       <U30C7>   KATAKANA LETTER DE
<To>       <U30C8>   KATAKANA LETTER TO
<Do>       <U30C9>   KATAKANA LETTER DO
<Na>       <U30CA>   KATAKANA LETTER NA
<Ni>       <U30CB>   KATAKANA LETTER NI
<Nu>       <U30CC>   KATAKANA LETTER NU
<Ne>       <U30CD>   KATAKANA LETTER NE
<No>       <U30CE>   KATAKANA LETTER NO
<Ha>       <U30CF>   KATAKANA LETTER HA
<Ba>       <U30D0>   KATAKANA LETTER BA
<Pa>       <U30D1>   KATAKANA LETTER PA
<Hi>       <U30D2>   KATAKANA LETTER HI
<Bi>       <U30D3>   KATAKANA LETTER BI
<Pi>       <U30D4>   KATAKANA LETTER PI
<Hu>       <U30D5>   KATAKANA LETTER HU
<Bu>       <U30D6>   KATAKANA LETTER BU
<Pu>       <U30D7>   KATAKANA LETTER PU
<He>       <U30D8>   KATAKANA LETTER HE
<Be>       <U30D9>   KATAKANA LETTER BE
<Pe>       <U30DA>   KATAKANA LETTER PE
<Ho>       <U30DB>   KATAKANA LETTER HO
<Bo>       <U30DC>   KATAKANA LETTER BO
<Po>       <U30DD>   KATAKANA LETTER PO
<Ma>       <U30DE>   KATAKANA LETTER MA
<Mi>       <U30DF>   KATAKANA LETTER MI
<Mu>       <U30E0>   KATAKANA LETTER MU
<Me>       <U30E1>   KATAKANA LETTER ME
<Mo>       <U30E2>   KATAKANA LETTER MO
<YA>       <U30E3>   KATAKANA LETTER SMALL YA
<Ya>       <U30E4>   KATAKANA LETTER YA
<YU>       <U30E5>   KATAKANA LETTER SMALL YU
<Yu>       <U30E6>   KATAKANA LETTER YU
<YO>       <U30E7>   KATAKANA LETTER SMALL YO
<Yo>       <U30E8>   KATAKANA LETTER YO
<Ra>       <U30E9>   KATAKANA LETTER RA
<Ri>       <U30EA>   KATAKANA LETTER RI
<Ru>       <U30EB>   KATAKANA LETTER RU
<Re>       <U30EC>   KATAKANA LETTER RE
<Ro>       <U30ED>   KATAKANA LETTER RO
<WA>       <U30EE>   KATAKANA LETTER SMALL WA
<Wa>       <U30EF>   KATAKANA LETTER WA
<Wi>       <U30F0>   KATAKANA LETTER WI
<We>       <U30F1>   KATAKANA LETTER WE
<Wo>       <U30F2>   KATAKANA LETTER WO
<N6>       <U30F3>   KATAKANA LETTER N
<Vu>       <U30F4>   KATAKANA LETTER VU
<KA>       <U30F5>   KATAKANA LETTER SMALL KA
<KE>       <U30F6>   KATAKANA LETTER SMALL KE
<Va>       <U30F7>   KATAKANA LETTER VA
<Vi>       <U30F8>   KATAKANA LETTER VI
<Ve>       <U30F9>   KATAKANA LETTER VE
<Vo>       <U30FA>   KATAKANA LETTER VO
<.6>       <U30FB>   KATAKANA MIDDLE DOT
<-6>       <U30FC>   KATAKANA-HIRAGANA PROLONGED SOUND MARK
```



```
<*6>       <U30FD>   KATAKANA ITERATION MARK
<+6>       <U30FE>   KATAKANA VOICED ITERATION MARK
<b4>       <U3105>   BOPOMOFO LETTER B
<p4>       <U3106>   BOPOMOFO LETTER P
<m4>       <U3107>   BOPOMOFO LETTER M
<f4>       <U3108>   BOPOMOFO LETTER F
<d4>       <U3109>   BOPOMOFO LETTER D
<t4>       <U310A>   BOPOMOFO LETTER T
<n4>       <U310B>   BOPOMOFO LETTER N
<l4>       <U310C>   BOPOMOFO LETTER L
<g4>       <U310D>   BOPOMOFO LETTER G
<k4>       <U310E>   BOPOMOFO LETTER K
<h4>       <U310F>   BOPOMOFO LETTER H
<j4>       <U3110>   BOPOMOFO LETTER J
<q4>       <U3111>   BOPOMOFO LETTER Q
<x4>       <U3112>   BOPOMOFO LETTER X
<zh>       <U3113>   BOPOMOFO LETTER ZH
<ch>       <U3114>   BOPOMOFO LETTER CH
<sh>       <U3115>   BOPOMOFO LETTER SH
<r4>       <U3116>   BOPOMOFO LETTER R
<z4>       <U3117>   BOPOMOFO LETTER Z
<c4>       <U3118>   BOPOMOFO LETTER C
<s4>       <U3119>   BOPOMOFO LETTER S
<a4>       <U311A>   BOPOMOFO LETTER A
<o4>       <U311B>   BOPOMOFO LETTER O
<e4>       <U311C>   BOPOMOFO LETTER E
<eh4>      <U311D>   BOPOMOFO LETTER EH
<ai>       <U311E>   BOPOMOFO LETTER AI
<ei>       <U311F>   BOPOMOFO LETTER EI
<au>       <U3120>   BOPOMOFO LETTER AU
<ou>       <U3121>   BOPOMOFO LETTER OU
<an>       <U3122>   BOPOMOFO LETTER AN
<en>       <U3123>   BOPOMOFO LETTER EN
<aN>       <U3124>   BOPOMOFO LETTER ANG
<eN>       <U3125>   BOPOMOFO LETTER ENG
<er>       <U3126>   BOPOMOFO LETTER ER
<i4>       <U3127>   BOPOMOFO LETTER I
<u4>       <U3128>   BOPOMOFO LETTER U
<iu>       <U3129>   BOPOMOFO LETTER IU
<v4>       <U312A>   BOPOMOFO LETTER V
<nG>       <U312B>   BOPOMOFO LETTER NG
<gn>       <U312C>   BOPOMOFO LETTER GN
<(JU)>     <U321C>   PARENTHESIZED HANGUL CIEUC U
<1c>       <U3220>   PARENTHESIZED IDEOGRAPH ONE
<2c>       <U3221>   PARENTHESIZED IDEOGRAPH TWO
<3c>       <U3222>   PARENTHESIZED IDEOGRAPH THREE
<4c>       <U3223>   PARENTHESIZED IDEOGRAPH FOUR
<5c>       <U3224>   PARENTHESIZED IDEOGRAPH FIVE
<6c>       <U3225>   PARENTHESIZED IDEOGRAPH SIX
<7c>       <U3226>   PARENTHESIZED IDEOGRAPH SEVEN
<8c>       <U3227>   PARENTHESIZED IDEOGRAPH EIGHT
<9c>       <U3228>   PARENTHESIZED IDEOGRAPH NINE
<10c>      <U3229>   PARENTHESIZED IDEOGRAPH TEN
<KSC>      <U327F>   KOREAN STANDARD SYMBOL
<am>       <U33C2>   SQUARE AM
<pm>       <U33D8>   SQUARE PM
<ff>       <UFB00>   LATIN SMALL LIGATURE FF
<fi>       <UFB01>   LATIN SMALL LIGATURE FI
<fl>       <UFB02>   LATIN SMALL LIGATURE FL
<ffi>      <UFB03>   LATIN SMALL LIGATURE FFI
<ffl>      <UFB04>   LATIN SMALL LIGATURE FFL
<St>       <UFB05>   LATIN SMALL LIGATURE LONG S T
<st>       <UFB06>   LATIN SMALL LIGATURE ST
<3+;>      <UFE7D>   ARABIC SHADDA MEDIAL FORM
<aM.>      <UFE82>   ARABIC LETTER ALEF WITH MADDA ABOVE FINAL FORM
<aH.>      <UFE84>   ARABIC LETTER ALEF WITH HAMZA ABOVE FINAL FORM
<ah.>      <UFE88>   ARABIC LETTER ALEF WITH HAMZA BELOW FINAL FORM
<a+->      <UFE8D>   ARABIC LETTER ALEF ISOLATED FORM
<a+.>      <UFE8E>   ARABIC LETTER ALEF FINAL FORM
<b+->      <UFE8F>   ARABIC LETTER BEH ISOLATED FORM
<b+.>      <UFE90>   ARABIC LETTER BEH FINAL FORM
<b+,>      <UFE91>   ARABIC LETTER BEH INITIAL FORM
<b+;>      <UFE92>   ARABIC LETTER BEH MEDIAL FORM
<tm->      <UFE93>   ARABIC LETTER TEH MARBUTA ISOLATED FORM
```



```
<tm.>      <UFE94>   ARABIC LETTER TEH MARBUTA FINAL FORM
<t+->      <UFE95>   ARABIC LETTER TEH ISOLATED FORM
<t+.>      <UFE96>   ARABIC LETTER TEH FINAL FORM
<t+,>      <UFE97>   ARABIC LETTER TEH INITIAL FORM
<t+;>      <UFE98>   ARABIC LETTER TEH MEDIAL FORM
<tk->      <UFE99>   ARABIC LETTER THEH ISOLATED FORM
<tk.>      <UFE9A>   ARABIC LETTER THEH FINAL FORM
<tk,>      <UFE9B>   ARABIC LETTER THEH INITIAL FORM
<tk;>      <UFE9C>   ARABIC LETTER THEH MEDIAL FORM
<g+->      <UFE9D>   ARABIC LETTER JEEM ISOLATED FORM
<g+.>      <UFE9E>   ARABIC LETTER JEEM FINAL FORM
<g+,>      <UFE9F>   ARABIC LETTER JEEM INITIAL FORM
<g+;>      <UFEA0>   ARABIC LETTER JEEM MEDIAL FORM
<hk->      <UFEA1>   ARABIC LETTER HAH ISOLATED FORM
<hk.>      <UFEA2>   ARABIC LETTER HAH FINAL FORM
<hk,>      <UFEA3>   ARABIC LETTER HAH INITIAL FORM
<hk;>      <UFEA4>   ARABIC LETTER HAH MEDIAL FORM
<x+->      <UFEA5>   ARABIC LETTER KHAH ISOLATED FORM
<x+.>      <UFEA6>   ARABIC LETTER KHAH FINAL FORM
<x+,>      <UFEA7>   ARABIC LETTER KHAH INITIAL FORM
<x+;>      <UFEA8>   ARABIC LETTER KHAH MEDIAL FORM
<d+->      <UFEA9>   ARABIC LETTER DAL ISOLATED FORM
<d+.>      <UFEAA>   ARABIC LETTER DAL FINAL FORM
<dk->      <UFEAB>   ARABIC LETTER THAL ISOLATED FORM
<dk.>      <UFEAC>   ARABIC LETTER THAL FINAL FORM
<r+->      <UFEAD>   ARABIC LETTER REH ISOLATED FORM
<r+.>      <UFEAE>   ARABIC LETTER REH FINAL FORM
<z+->      <UFEAF>   ARABIC LETTER ZAIN ISOLATED FORM
<z+.>      <UFEB0>   ARABIC LETTER ZAIN FINAL FORM
<s+->      <UFEB1>   ARABIC LETTER SEEN ISOLATED FORM
<s+.>      <UFEB2>   ARABIC LETTER SEEN FINAL FORM
<s+,>      <UFEB3>   ARABIC LETTER SEEN INITIAL FORM
<s+;>      <UFEB4>   ARABIC LETTER SEEN MEDIAL FORM
<sn->      <UFEB5>   ARABIC LETTER SHEEN ISOLATED FORM
<sn.>      <UFEB6>   ARABIC LETTER SHEEN FINAL FORM
<sn,>      <UFEB7>   ARABIC LETTER SHEEN INITIAL FORM
<sn;>      <UFEB8>   ARABIC LETTER SHEEN MEDIAL FORM
<c+->      <UFEB9>   ARABIC LETTER SAD ISOLATED FORM
<c+.>      <UFEBA>   ARABIC LETTER SAD FINAL FORM
<c+,>      <UFEBB>   ARABIC LETTER SAD INITIAL FORM
<c+;>      <UFEBC>   ARABIC LETTER SAD MEDIAL FORM
<dd->      <UFEBD>   ARABIC LETTER DAD ISOLATED FORM
<dd.>      <UFEBE>   ARABIC LETTER DAD FINAL FORM
<dd,>      <UFEBF>   ARABIC LETTER DAD INITIAL FORM
<dd;>      <UFEC0>   ARABIC LETTER DAD MEDIAL FORM
<tj->      <UFEC1>   ARABIC LETTER TAH ISOLATED FORM
<tj.>      <UFEC2>   ARABIC LETTER TAH FINAL FORM
<tj,>      <UFEC3>   ARABIC LETTER TAH INITIAL FORM
<tj;>      <UFEC4>   ARABIC LETTER TAH MEDIAL FORM
<zH->      <UFEC5>   ARABIC LETTER ZAH ISOLATED FORM
<zH.>      <UFEC6>   ARABIC LETTER ZAH FINAL FORM
<zH,>      <UFEC7>   ARABIC LETTER ZAH INITIAL FORM
<zH;>      <UFEC8>   ARABIC LETTER ZAH MEDIAL FORM
<e+->      <UFEC9>   ARABIC LETTER AIN ISOLATED FORM
<e+.>      <UFECA>   ARABIC LETTER AIN FINAL FORM
<e+,>      <UFECB>   ARABIC LETTER AIN INITIAL FORM
<e+;>      <UFECC>   ARABIC LETTER AIN MEDIAL FORM
<i+->      <UFECD>   ARABIC LETTER GHAIN ISOLATED FORM
<i+.>      <UFECE>   ARABIC LETTER GHAIN FINAL FORM
<i+,>      <UFECF>   ARABIC LETTER GHAIN INITIAL FORM
<i+;>      <UFED0>   ARABIC LETTER GHAIN MEDIAL FORM
<f+->      <UFED1>   ARABIC LETTER FEH ISOLATED FORM
<f+.>      <UFED2>   ARABIC LETTER FEH FINAL FORM
<f+,>      <UFED3>   ARABIC LETTER FEH INITIAL FORM
<f+;>      <UFED4>   ARABIC LETTER FEH MEDIAL FORM
<q+->      <UFED5>   ARABIC LETTER QAF ISOLATED FORM
<q+.>      <UFED6>   ARABIC LETTER QAF FINAL FORM
<q+,>      <UFED7>   ARABIC LETTER QAF INITIAL FORM
<q+;>      <UFED8>   ARABIC LETTER QAF MEDIAL FORM
<k+->      <UFED9>   ARABIC LETTER KAF ISOLATED FORM
<k+.>      <UFEDA>   ARABIC LETTER KAF FINAL FORM
<k+,>      <UFEDB>   ARABIC LETTER KAF INITIAL FORM
<k+;>      <UFEDC>   ARABIC LETTER KAF MEDIAL FORM
<l+->      <UFEDD>   ARABIC LETTER LAM ISOLATED FORM
```

```
<l+.>      <UFEDE>   ARABIC LETTER LAM FINAL FORM
<l+,>      <UFEDF>   ARABIC LETTER LAM INITIAL FORM
<l+;>      <UFEE0>   ARABIC LETTER LAM MEDIAL FORM
<m+->      <UFEE1>   ARABIC LETTER MEEM ISOLATED FORM
<m+.>      <UFEE2>   ARABIC LETTER MEEM FINAL FORM
<m+,>      <UFEE3>   ARABIC LETTER MEEM INITIAL FORM
<m+;>      <UFEE4>   ARABIC LETTER MEEM MEDIAL FORM
<n+->      <UFEE5>   ARABIC LETTER NOON ISOLATED FORM
<n+.>      <UFEE6>   ARABIC LETTER NOON FINAL FORM
<n+,>      <UFEE7>   ARABIC LETTER NOON INITIAL FORM
<n+;>      <UFEE8>   ARABIC LETTER NOON MEDIAL FORM
<h+->      <UFEE9>   ARABIC LETTER HEH ISOLATED FORM
<h+.>      <UFEEA>   ARABIC LETTER HEH FINAL FORM
<h+,>      <UFEEB>   ARABIC LETTER HEH INITIAL FORM
<h+;>      <UFEEC>   ARABIC LETTER HEH MEDIAL FORM
<w+->      <UFEED>   ARABIC LETTER WAW ISOLATED FORM
<w+.>      <UFEEE>   ARABIC LETTER WAW FINAL FORM
<j+->      <UFEEF>   ARABIC LETTER ALEF MAKSURA ISOLATED FORM
<j+.>      <UFEF0>   ARABIC LETTER ALEF MAKSURA FINAL FORM
<y+->      <UFEF1>   ARABIC LETTER YEH ISOLATED FORM
<y+.>      <UFEF2>   ARABIC LETTER YEH FINAL FORM
<y+,>      <UFEF3>   ARABIC LETTER YEH INITIAL FORM
<y+;>      <UFEF4>   ARABIC LETTER YEH MEDIAL FORM
<lM->      <UFEF5>   ARABIC LIGATURE LAM WITH ALEF WITH MADDA ABOVE ISOLATED FORM
<lM.>      <UFEF6>   ARABIC LIGATURE LAM WITH ALEF WITH MADDA ABOVE FINAL FORM
<lH->      <UFEF7>   ARABIC LIGATURE LAM WITH ALEF WITH HAMZA ABOVE ISOLATED FORM
<lH.>      <UFEF8>   ARABIC LIGATURE LAM WITH ALEF WITH HAMZA ABOVE FINAL FORM
<lh->      <UFEF9>   ARABIC LIGATURE LAM WITH ALEF WITH HAMZA BELOW ISOLATED FORM
<lh.>      <UFEFA>   ARABIC LIGATURE LAM WITH ALEF WITH HAMZA BELOW FINAL FORM
<la->      <UFEFB>   ARABIC LIGATURE LAM WITH ALEF ISOLATED FORM
<la.>      <UFEFC>   ARABIC LIGATURE LAM WITH ALEF FINAL FORM
<H->       <U0023>   NUMBER SIGN
<!S>       <U0024>   DOLLAR SIGN
<@>        <U0040>   COMMERCIAL AT
<Oa>       <U0040>   COMMERCIAL AT
<!C>       <U00A2>   CENT SIGN
<L->       <U00A3>   POUND SIGN
<Xo>       <U00A4>   CURRENCY SIGN
<Y->       <U00A5>   YEN SIGN
<!B>       <U00A6>   BROKEN BAR
<So>       <U00A7>   SECTION SIGN
<7!>       <U00AC>   NOT SIGN
<9I>       <U00B6>   PILCROW SIGN
<_->       <U2500>   BOX DRAWINGS LIGHT HORIZONTAL
<_=>       <U2501>   BOX DRAWINGS HEAVY HORIZONTAL
<_!>       <U2502>   BOX DRAWINGS LIGHT VERTICAL
<_V/>>     <U250C>   BOX DRAWINGS LIGHT DOWN AND RIGHT
<_V<w>     <U2510>   BOX DRAWINGS LIGHT DOWN AND LEFT
<_A/>>     <U2514>   BOX DRAWINGS LIGHT UP AND RIGHT
<_A<>      <U2518>   BOX DRAWINGS LIGHT UP AND LEFT
<_!/>>     <U251C>   BOX DRAWINGS LIGHT VERTICAL AND RIGHT
<_!<>      <U2524>   BOX DRAWINGS LIGHT VERTICAL AND LEFT
<_V->      <U252C>   BOX DRAWINGS LIGHT DOWN AND HORIZONTAL
<_-A>      <U2534>   BOX DRAWINGS LIGHT UP AND HORIZONTAL
<_!->      <U253C>   BOX DRAWINGS LIGHT VERTICAL AND HORIZONTAL
<_/>//>    <U2571>   BOX DRAWINGS LIGHT DIAGONAL UPPER RIGHT TO LOWER LEFT
<_<\>      <U2572>   BOX DRAWINGS LIGHT DIAGONAL UPPER LEFT TO LOWER RIGHT
<_./>//>   <U25E2>   BLACK LOWER RIGHT TRIANGLE
<_.<\>     <U25E3>   BLACK LOWER LEFT TRIANGLE
<_d!>      <U266A>   EIGHTH NOTE
```

7. FUNCTIONALITY

Functionality to access the information described in this International Standard is described in ISO/IEC 9899 – Programming Language C. In addition the following functions are specified:

7.1 The “strpcoll” function

Synopsis

```
#include <string.h>
int strpcoll(const char *s1, const char *s2, int p);
```

Description

The strpcoll function compares the string pointed to by s1 to the string pointed to by s2, both interpreted as appropriate to the LC_COLLATE category of the current FDCC-set, and to the precision of p.

Returns

The strpcoll function returns an integer greater than, equal to, or less than zero, accordingly as the string pointed to by s1 is greater than, equal to, or less than the string pointed to by s2, given the precision p, when both are interpreted as appropriate to the current FDCC-set. The precision p is the level of the collation data that will be used, on alphabetic characters, precision=1 will normally regard all versions, including upper case, lowercase and accented versions of a letter as equal; and precision=2 will normally regard upper case and lower case versions of an accented letter as equal, both in composed and decomposed forms; precision=3 would normally distinction between composed and decomposed forms of a letter.

7.2 The “setmedia” function

Synopsis

```
int setmedia(int io, int media, int allow);
```

Description

The setmedia function sets the message interfaces to be used. io gives input or output; input=0, output=1. media gives the media type: text=1, voice=2, gestures=3, selection=4. allow specifies if the combination of io and media will be allowed or not, allow=1, and deny=0.

Returns

The setmedia function returns a 0 if the function executes without error, and a positive non-zero error code if the function executes with error.

7.3 String, encoding, repertoire, and locale data types

As basic string handling is dependent on the user's preferences (as given via the string), encoding,

repertoire, and locale data types are described together here.

7.3.1 String data type

The string handling APIs defined in this document operate on an internal representation of character strings, which are arrays of characters, A void string is indicated by the implementation-defined value NIL. The string data type is defined by:

```
#define string *wchar_t
```

7.3.2 Encoding data type

The "encoding" data type holds data necessary to convert to and from an external encoding and the internal string representation. This includes mapping of coded characters to the internal repertoire, how to shift between subencodings such as via ISO 2022 techniques, or representation via symbolic character names identified via introducing sequences, and state information. The enocding data type is defined by:

```
struct encoding {
        string encodingname;
        // other things ???
}
```

NOTE The encoding definition is closely related to the "charmap" specification of POSIX and TR 14652, the "charset" definition in the Internet MIME specification, and newer developments for the C and C++ programming languages.

7.3.2.1 int newencoding(const string encodingname,encoding enc)

The "newencoding" API creates an encoding object with the necessary space to hold all information necessary to convert between the encoding and the internal string representation. The "newencoding" API sets default values, including the "line_terminator" to being the characters "CR""LF", the "invalid_char" to being the character "SUB", the "symbolic_char_introducer" to being "NUL" (not valid), the "sub_encoding_change" API, the "get_symbolic_char_name" API and the "put_symbolic_char_name" API to be the null API, and the "input_state" and the "output_state" variables to be the initial state .

The "encodingname" is an implementation defined string with the following characterisics:

An initial string of "std/" refers to the charmaps registered in the international cultural register, ISO/IEC 15897.

If the specified encoding is syntactically valid and supported by the implementation, the "newencoding" API allocates memory for the new object and returns a pointer to the object in the parameter "enc". It is the application's responsibility to free this memory with a call to the "freeencoding" API when the struct is no longer needed. If the API fails for any reason, the contents of "enc" is undefined.

The "newencoding" API returns one of the following values:

0 -LC_SUCCESS - The API call was successful

1 - LC_NOTSUPPORTED - The encoding is not supported by the current system.

2 - LC_NOMEMORY - there was insufficient memory to perform the API

3 -LC_INVALID - The specified encoding is invalid

7.3.2.2 int freeencoding(encoding enc)

The "freeencoding" API frees the memory occupied by the encoding "enc". It returns a zero if the operation is succesful, and a -1 otherwise.

7.3.2.4 int setencint(encoding enc,const string param,int val)

iThe "setencint" API sets a specific parameter as specified in the string "param" of the encoding specification to a specific integer value as specified in "val". The defined values for "param" are:
(to be described)
It returns a zero if the operation is succesful, and a 1 otherwise.

7.3.2.4 int setencbytes(encoding enc,const string param,const char *val,int len)

The "setencbytes" API sets a specific parameter as specified in the string "param" of the encoding specification to a specific multibyte value as specified in "val" with the length "len" bytes. The defined values for "param" are:
"line_terminator"
"invalid_char"
"symbolic_char_introducer"

7.3.2.5 int setencproc(encoding enc, const string param, int val())

The "setencproc" API sets a specific parameter as specified in the string "param" of the encoding specification to a specific API value as specified in "val" . The defined values for "param" are:
"sub_encoding_change" - procedure subec()
"get_symbolic_char_name" - procedure gscn(c, p , len)
"put_symbolic_char_name" - procedure pscn(c, p, len)

7.3.2.6 int gscn(c, p, len)

The application defined "get_symbolic_char_name" API is called by the "bytes2string" API when the character sequence in "symbolic_char_introducer" is met in the input octet sequence. It gets a pointer "p" to the first octet after the "symbolic_char_introducer" and determines whether there is a symbolic character according to the application APIs definitions, with or without a terminator sequence, within "len" octets after the "p" pointer. If successful the API returns the found character in the internal string representation in "c" and the pointer "p" to the first octet after the symbolic character, including the possible terminator sequence. The application defined API returns:
0 if succesful
1 if it could not recognise a symbolic character within the "len" octets. "p" is not changed.
2 if the octet sequence is invalid according to the rules of the application. "p" is not changed.

7.3.2.7 int pscn(c, p, len)

The application defined "put_symbolic_char_name" API is called by the "string2bytes" API when a character is not present in the external encoding. It gets a pointer "p" to the next octet to be written in the sequence of octets and determines whether there is room to put a symbolic character according to the application APIs definitions, with the "symbolic_char_introducer" value and with or without a terminator sequence, within "len" octets after the "p" pointer. If successful, the API returns "p", a pointer, to the first octet after the symbolic character written, including the possible terminator sequence. The application defined API returns:
0 if successful
1 if the API was not able to write the symbolic character within "len" octets. The pointer "p" is not changed.
2 if the API had no means of writing the character "c", The pointer "p" is not changed.

7.3.3 Repertoire data type

The "repertoire" data type holds data necessary for the "stringtrans" transliteration API.

7.3.3.1 int newrepertoire(const string repertoirename,repertoire rep)

The "newrepertoire" API creates a repertoire object with the necessary space to hold all information necessary. The "repertoirename" is an implementation defined string with the following characteristics: An initial string of "std/" refers to repertoiremaps registered in the international cultural register, ISO/IEC 15897. If the specified repertoire is valid and supported, the "newrepertoire" API allocates memory for the new object and returns a pointer to the object in "rep". It is the application's responsibility to free this memory with a call to the "freerepertoire" API when the object is no longer needed. If the API fails for any reason, the contents of "rep" is undefined.

The "newrepertoire" API returns one of the following values:

0 - The API call was successful
1 - The repertoire is not supported by the current system.
2 - There was insufficient memory to perform the API
3 - The specified repertoire is invalid

7.3.3.2 int freerepertoire(repertoire rep)

The "freerepertoire" API frees the memory occupied by the repertoire "rep". It returns 0 if the operation is successful, and 1 otherwise.

7.3.3.3 int enc2repertoire(encoding enc,repertoire rep)

The "enc2repertoire" API generates a repertoire object with a repertoire corresponding to the character repertoire of the encoding "enc". If the API is successful, it returns the repertoire object in "rep". It has the same return values as the "newrepertoire" API.

7.3.4 Locale data type

The "locale" data type is a pointer to a struct with a number of variables capable of holding information sufficient to service all language-dependent internationalization services. The "locale" data type has provisions to affect groups of functionalities in categories, which are:

1 NULL
2 LC_ALL
3 LC_IDENTIFICATION
4 LC_COLLATE
5 LC_CTYPE
6 LC_MONETARY
7 LC_NUMERIC
8 LC_TIME
9 LC_MESSAGES
10 LC_XLITERATE
11 LC_NAME
12 LC_ADDRESS
13 LC_TELEPHONE

The category LC_ALL denotes all of the other non-void categories.

The category NULL denotes a void category.

The "locale" data type includes the following variables (which are further described in this document:)

LC_MONETARY values:

int_curr_symb: string.
currency_symbol: string.
mon_deccimal_point: string.
mon_thousands_sep: string.
mon_grouping: string
positive_sign: string.
negative_sign: string.
int_frac_digits: integer.
frac_digits: integer.
p_cs_precedes: integer
p_sep_by_space: integer.
n_cs_precedes: integer
n_sep_by_space: integer
p_sign_posn: integer
n_sign_posn: integer

LC_NUMERIC values:

decimal_point: string
thousands_sep: string
grouping: array of integers

LC_TIME values:

abday: array (1,7) of string
day: array (1,7) of string
abmon: array (1,13) of string
mon: array (1,13) of string
d_t_fmt: string
d_fmt: string
t_fmt: string
am_pm: string
t_fmt_ampm: string
era: string
era_year: string
era_d_fmt: string
alt_digits: array (1,100) of string

LC_MESSAGES values:

yesexpr: string
noexpr: string

7.3.4.1 int newlocale(int category_mask, const string localename,locale lc)

The "newlocale" API creates a locale struct with all the necessary information to perform the language-sensitive operations of internationalization APIs accepting an argument of the type "locale". If the API is successful, all categories of the locale object are created and initialized. Any categories in the locale identified by "localename" are initalized to the i18n locale.

The "localename" is an implementation-defined value with the following characteristics:
- An initial string of "std/" refers to the locales registered in the international cultural register, ISO/IEC 15897.

If the specified locale is valid and supported, the "newlocale" API allocates memory for the new object and returns a pointer to the object in "lc". It is the application's responsibility to free this memory with a call to the "freelocale" API when the object is no longer needed. If the API fails for any reason, the contents of "lc" is undefined.

The "newlocale" API returns one of the following values:

0 - LC_SUCCESS - The API call was successful.

1 - LC_INCOMPLETE - The specified locale has been created, but the locale object contains one or more categories that were initialized to the i18n locale because the "localename" did not identify a value for that category.

2 - LC_NOTSUPPORTED - The locale is not supported by the current system.

3 - LC_NOMEMORY - there was insufficient memory to perform the API.

4 - LC_INVALID - The specified locale is invalid.

7.3.4.2 int freelocale(locale lc)

The "freelocale" API frees the memory occupied by the locale "lc". It returns 0 if the operation is succesful, and 1 otherwise.

7.3.4.3 int modifylocale(const int category,const string localename,locale lc)

The "modifylocale" API modifies the values of the locale object "lc" parameter relating to the category "category" and with values as specified in "localename". "category" takes values as defined in 5.4 and "localename" is defined as for the "newlocale" API. The return value is as for the "newlocale" API.

7.3.4.4 int intlocaleinfo(const int category,const string keywordname,locale lc)

The "intlocaleinfo" API gets the integer value of the keyword "keywordname" of the locale object "lc" relating to the category "category". "category" takes values as defined in 5.4.The return value is the integer value of the keyword.

7.3.4.5 string stringlocaleinfo(const int category,const string keywordname,locale lc)

The "stringlocaleinfo" API gets the string value of the keyword "keywordname" of the locale object "lc" relating to the category "category". "category" takes values as defined in 5.4.The return value is the string value of the keyword.

7.3.5 Character handling

The character handling APIs behave according to the LC_CTYPE category of the locale parameter for the individual APIs.

7.3.5.1 int istype(wchar_t c,const string c_type,const locale lc)

The "istype" API returns 1 if the character "c" is in the type "c_type", else 0.
"c_type" can have the following values:
alnum, alpha, cntrl, digit, graph, lower, print, punct, space, blank, upper, xdigit

7.3.5.2 int tolowers(string s1,const string s2,const locale lc)

The "tolowers" API returns in string "s1" all characters in the string "s2" converted to the corresponding lowercase characters with conversion rules given by the locale "lc". The API returns the number of resulting characters in "s1".

7.3.5.3 int touppers(string s1,const string s2,const locale lc)

The "touppers" API returns in string "s1" all characters in the string "s2" converted to the corresponding uppercase characters with conversion rules given by the locale "lc". The API returns the number of resulting charactes in "s1".

7.3.5.4 int stringtrans(transtype, maxlen,string s1,const string s2, rep)

The "stringtrans" API transforms string "s2" into string "s1" given the transformation specifications as noted below.

Values for the "transtype" parameter are

1 - as for the "tolowers" API

2 - as for the "touppers" API

3 - transliterate the string "s2" into the string "s1" (for example using for each character the first "transform" specification of ISO/IEC TR 14652) that is using the repertoire of "rep" and has at most "maxlen" characters as the transliteration. If the "s1" string is to be exceeded, or there is no valid transliteration, the API returns -1. Otherwise it returns the resulting number of characters of "s1".

7.3.6 String comparison

The string comparison APIs behave according to the LC_COLLATE category of the locale parameter for the individual APIs.

7.3.6.1 int strcoll_l(const string s1,const string s2,locale lc)

int strncoll_l(const string s1,const string s2, n,locale lc)

The "strcoll_l" API compares the two strings "s1" and "s2" with regards to the collating specifications of the locale "lc".

The "strncoll_l" API compares at most "n" characters of the two strings "s1" and "s2" with regards to the collating specifications of the locale "lc".

Both the "strcoll_l" and "strncoll_l" APIs returns -1 if "s1" < "s2", 0 if "s1" == "s2" and 1 if "s1" > "s2" .

7.3.6.2 int strxfrm_l(const string s1,const string s2,locale lc)

The "stringxfrm" API converts the character string "s2" using the locale "lc" and to the precision in "precision"as defined in 7.2, to an internal representation in "s1" suitable for comparison via a binary comparison API (in C this may be strcmp()).

7.3.6.3 int stringcoll(const string s1,const string s2,int precision,locale lc)

7.3.6.4 int stringncoll(const string s1,const string s2,int precision,int n,locale lc)

The "stringcoll" API compares the two strings "s1" and "s2" with regards to the collating specifications of the locale "lc" and to the precision in "precision".

The "stringncoll" API compares at most "n" characters of the two strings "s1" and "s2" with regards to the collating specifications of the locale "lc" and to the precision in "precision".

The "precision" indicates to what level of preciseness the string comparison is done. "precision" may have the following values:

0 - all levels
1 - only to level 1 - CASE_AND_ACCENT_INSENSITITVE
2 - only to level 2 - CASE_INSENSITIVE
3 - only to level 3 - IGNORE_SPECIALS
4 - only to level 4 - EXACT_MATCHING

Both the "stringcoll" and "stringncoll" APIs returns -1 if "s1" < "s2", 0 if "s1" == "s2" and 1 if "s1" > "s2" .

7.3.6.5 int stringxfrm(const string s1,const string s2,int precision,locale lc)

The "stringxfrm" API converts the character string "s2" using the locale "lc" and to the precision in "precision" as defined in 7.2, to an internal representation in "s1" suitable for comparison via a binary comparison API (in C this may be strcmp()).

7.3.7 Message formatting

The message formatting APIs behave according to the LC_MESSAGES category of the locale parameter for the individual APIs.

7.3.7.1 string stringget(const string msgtag,const string textdomain,locale lc)

The "stringget" API gets the message with the tag "msgtag" in the current LC-MESSAGES part of the "lc" locale with respect to the "textdomain" set of messages. If not found or the locale is invalid and no "msgtag" is found in the default locale, then "msgtag" is returned.

7.3.8 Conversion between string and other data types

7.3.8.1 int string2int_l(string s,locale lc)

The "string2int_l" API converts a string to an integer, with respect to the locale "lc".

7.3.8.2 string int2string_l(int i,locale lc)

The "int2string_l" API creates a string with the necessary length and returns the string with an integer formatted in characters, according to the locale "lc". If there is not enough memory to create a new string, the API returns the void string.

7.3.8.3 double string2real_l(string s,locale lc)

The "string2real_l" API converts a string to a real value, using information about thousands and decimal separators from the locale "lc". If there is not enough memory to create a new string, the API returns the void string.

7.3.8.4 string real2string_l(double r,locale lc)

The "real2string_l" API formats a real value into a string, with decimal and thousands separators

as given in the "lc" locale. Returns a string with the necessary length, or if memory is not available it returns the empty string.

7.3.8.5 int bytes2string_e(string s,char* p,int len,encoding enc)

The "bytes2string_e" API converts "len" octets from the multibyte value "p" in the encoding "enc" to the string "s", and with the conversion input_state as recorded in "enc". The conversion stops earlier in two cases: if the next character to be stored in the string "s" would exceed the length of "s", or if there is a sequence not corresponding to a recognizable character in the input sequence of octets, possibly after calling an application-defined "get_symbolic_char" API.

If the API stops without having converted "len" octets, the API returns the negative to the number of octets converted. Otherwise it returns the number of internal characters converted (ie. the last index in the string "s" for characters converted).

7.3.8.6 int string2bytes_e(char* p,string s,int len,encoding enc)

The "string2bytes_e" API converts a string "s" into a sequence of corresponding octets of "p" in the encoding "enc", and beginning in the output_state recorded in "enc". The conversion continues up to the lenght of the string "s". The conversion stops earlier in two cases: when a code is reached that does not correspond to a valid representation in the sequence of octets, and either no "invalid_char" value or "put_symbolic_char" API is defined, or the application defined "put_symbolic_char" API returns with a value 2; or the next octet would exceed the limit of "len" total octets to be stored in the multibyte "p" variable.

The API returns the negative index of the character in question if the conversion stops because it could not convert a character in the string to octets. Otherwise it returns the number of octets in the resulting sequence of octets.

7.3.8.7 int time2string_l(string s,const string format,const struct tm *timeptr,locale lc)

The "time2string_l" API returns a string "s" formatted according to the format in "format" of the time value in "timeptr", according to the local conventions in the locale "lc". The "format" string is specified in IS 9945 (with extensions as described in this document) as the "d_t_fmt" specification.

7.3.8.8 int string2time_l(const struct tm *time,string s,locale lc)

The "string2time_l" API returns a binary time in "time", scanned from the string "s" according to the locale conventions in the locale "lc".

NOTE: This specification needs more work. The C++ standard is the only standard having provisions for this, but is very weak on the subject.

7.3.8.9 int money2string_l(string s,const string format,const double amount,const struct tm *timeptr ,locale lc)

API "money2string_l" returns in parameter "s" a string formatted according to the format in "format" of the money value "amount". The formatting is done with respect to the locale "lc" at the time given in "time". The return value is the number of characters formatted, or -1 if an error occurred.

The parameter "format" is a string that consist of characters that shall be transfered to the output string "s" literally, and formatting specifications that specifies how the money value "amount" is to be formatted.

A formatting specification consist of the following sequence:
- a "%" character
- optional flags
- optional field width
- optional left precision
- optional right position
- the formatting character to specify which formatting to perform.

Flags are given with special characters, width and precision information is given with decimal digits, and formating characters are given with latin letters or the character "%".

The flags are:

"=f" an "=" followed by a single character which is used as the numeric fill character. No restriction is made on the representation of this single character. The default numeric fill character is the SPACE character. This flag does not affect the field width filling which always uses the SPACE character. This flag is ignored unless a left precision (se below) is specified.

"^" Do not use the grouping characters when formatting the amount. The default is to insert the grouping characters if defined in the locale "lc".

"+" or "(" Specify the style of representing positive and negative amounts. Only one of "+" or "(" may be specified. If "+" is specified, the equivalent of "+" and "-" are used from the locale "lc". If "(" is specified, negative amounts are enclosed within parenthesis. If neither flag is specified, the "+" style is used.

"!" Suppersses the currency symbol from the output conversion.

"-" Specify the alignment. If this flag is present all fields are left-justified (padded to the right) rather than right-justified.

Field Width

w A string of decimal digits specifying the minimum field width in characters in which the result of the conversion is right-justified (or left-justified if the "-" flag is specified) The default is 0.

Left Precision

"#n" A "#" followed by a string of decimal digits specifying a maximum number of digits expected to be formatted to the left of the radix character. This option can be used to keep the formatted output from several calls to API "money2string" aligned in the same coloumns. It can also be used to fill unused positions with a special character specified with the "=f" flag, as in $***123.45. If more than "n" positions are required, this formatting specification is ignored. Digit positions in excess of those actually required are filled with the numeric fill character, see the "=f" flag above.

If grouping has not been suppressed with the "^" flag, and it is defined for the locale "lc", grouping separators are inserted before the fill characters (if any) are added. Grouping separators are not applied to fill characters, even if the fill character is a digit.

To ensure alignment, any characters appearing before or after the number in the formatted output such as currency or sign, symbols are padded as neccessary with SPACE characters to make their positive or negative formats an equal length.

Right precision

".p" A "." followed by a string of decimal digits specifying the number after the radix character. If the value of the right precision is 0, no radix character appears. If a right precision is not included, the value specified in the "lc" locale is used. It is recommended to normally use the value from the locale. The amount being formatted is rounded to the specified number of digits prior to formatting.

Formatting characters:

The formatting characters and their meanings are:

"d" The following characters and up to any corresponding "%d" or the end of the formatting

string are only interpreted if there is a second currency in the "lc" locale for the time "t". The amount "a" is converted according to the "conversion_rate" of the locale "lc" and formatted according to any following formatting characters. No argument is converted. There shall be no flags, nor width or precision parameters, just "%%" is allowed.

"i" The type money argument is formatted acording to the "lc" locales's international currency format.

"n" The type money argument is formatted acording to the "lc" locales's national currency format.

"%" Convert to a "%"; no argument is converted. There shall be no flags, nor width or precision parameters, just "%%" is allowed.

7.3.8.10 int name2string_l(string s,const string format,const string name,locale lc)

The API “name2string_l” formats a set of personal name information as given in "name" to a string "s" according to the format in"format" and to the locale given in "lc". The format specification is unspecified (but a description may be found in TR 14652 for the keyword "name_fmt") if this is the empty string, the format specified in the "name_fmt" keyword of the locale in "lc" is used. The return value is the number of characters in the resulting string "s" or -1 if the supplied string "s" had indufficient length to hold the result.

The namerecord shall contain the following strings, which may each be empty:

family - family names, corresponding to the %f and %F escape sequence
given - first given name
giveninit - initial of given name
middle - middle names
middleinit - middle initials
shortname - a shorter name, eg. "Bill"
profession - the profession title
salutation - common salutation, like "Mr."
intsalut - a string with a digit in the range 1 to 5 for salutation

Similar strings with a "r" prepended to the name shall be present to hold Romanized information on the above items.

7.3.8.11 int address2string_l(string s,const string format,const string address,locale lc)

API “address2string_l” formats a set of address information as given in "address" to a string "s" according to the format in"format" and to the locale given in "lc". The format specification is unspecified, (but a description may be found in TR 14652 for the keyword "postal_fmt"); if this is the empty string, the format specified in the "postal_fmt" keyword of the locale in "lc" is used. The return value is the number of characters in the resulting string "s" - or -1 if an error occurred.

The addressrecord shall contain the following strings (with corresponding escape sequences given in parentheses), which may each be empty:

co - C/o address (%a)
firm - firm name (%f)
department - department name (%d)
building - building name (%b)
streetblock - street or block name (%s)
house - house number or designation (%h)
room - room number or designation (%r)
floor - floor number (%e)

village - village name (%v)
town - town or city name (%T)
countrycode - country designation or code (%C)
zip - zip or postal code (%z)
country- country name (%c)

Similar strings with a "r" prepended to the name shall be present to hold Romanized information on the above items.

7.3.8.12 int teldom2string_l(string s,const string format,const string telephone,locale lc)

API teldom2string_l formats for domestic use a telephone number as given in "telephone" to a string "s" according to the format in"format" and to the locale given in "lc". The format specification is unspecified (but a description may be found in TR 14652 for the keyword "tel_dom_fmt"); if this is the empty string, the format specified in the "tel_dom_fmt" keyword of the locale in "lc" is used. The return value is the number of characters in the resulting string "s" - or -1 if an error occurred.

7.3.8.13 int telint2string_l(string s,const string format,const string telephone,locale lc)

The API telint2string_l formats for international use a telephone number as given in "telephone" to a string "s" according to the format in"format" and to the locale given in "lc". The format specification is unspecified (but a description may be found in TR 14652 for the keyword "tel_int_fmt"); if this is the empty string, the format specified in the "tel_int_fmt" keyword of the locale in "lc" is used. The return value is the number of characters in the resulting string "s" - or -1 if an error occurred.

7.3.9 Utilities

Utilities are APIs that provide an interface at runtime, as a program.

localedef

localedef [-c] [-f charmap [-F char-repertoire]] [-i locale-source [-I locale-repertoire]] localename

The "localedef" utility shall convert source definitions for locale or FDCC-set categories into a format usable by the APIs and utilities whose operational behaviour is determined by the locale.

The utility shall read source definitions for one or more categories from the file named in the "-i" option (if specified) or from the standard input.

The "localename" identifies the target locale.

The following options shall be supported by the implementation:

-c Create permanent output even if warning messages have been issued.

-f charmap Specify the pathname of a file containing a mapping of character symbols and collating element symbols to actual character encoding. The format of the charmap is described in IS 9945 (with possible extentions described in TR 14652). This option shall be specified if symbolic names (other than collating symbols defined in the input locale or FDCC-set) are used. If the "-f" option is not present, an implementation-defined character mapping is used.

-F char-repertoire The pathname of a file containing a repertoiremap describing mapping between character symbols used in the charmap and IS 10646 characters.

-i locale-source The pathname of a file containing the source defitions of the categories. If this option is not present, shource definitions shall be read from standard input. The format of the

FDCC-set and locale is described in IS 9945 (with possible extensions described in TR 14652).

-I locale-repertoire The pathname of a file containing a repertoiremap describing mapping between character symbols used in the locale and IS 10646 characters.

The -F and -I options need only be specified if the use of character symbols differ between the locale-source and charmap.

The following operand shall be supported by the implementation

localename Identifies the output locale. If the name contains one or more <solidus> characters, "localename" shall be interpreted as a pathname where the created locale definiton(s) shall be stored. If "localename" does not contain any <solidus> characters, the interpretation of the name is implementation defined, and the locale shall be public. This capability may be restricted to users with appropriate privileges.

The utility shall report all categories successfully processed, in an unspecified format

The format of the created output file is unspecified.

The utility shall exit with one of the follwing values:

0 No errors occurred and the output files were sucessfully created

1 Warnings occurred and the output fileswere successfully created

2 The locale specifications exceeded implementation limits, or the charmap used was not supported by the implementation, and no output files were created.

>3 Warnings or errors occurred and no output files were created.

Consequence of errors

If an error is detected, no permanent output shall be created.

If warnings occur, permanent output shall be created if the "-c" option was specified. The following conditions shall cause warning messages to be issued:

- If a symbolic name used in the LC_CTYPE or LC_COLLATE categories cannot be matched to a corresponding symbolic name in the "charmap" (for other catergories, this shall be an error condition). The match is true if the symbolic name is found both in the source locale and in the charmap; or if a locale-repertoire and char-repertoire file is specified, the match is true if there exist a symbolic name in each of the repertoiremaps that match to the same character in IS 10646.

- If the number of operands to the "order_start" keyword exceeds the COLL_WEIGHTS_MAX limit

- If optional keywords not supported by the implementation are present in the source.

Other implementation-defined conditions may also cause warnings.

8. MESSAGES FORMAT

A file format to specify messages with an identifying specification, and a specification for the message in the context of the LC_MESSAGES category of the FDCC-set is specified in this clause. The format will allow for different media for the message, such as text, voice and gestures, and will allow for input and output. The specification is built on the gettext specification common in POSIX and C environments, and also used with other programming languages and systems. Other specification formats are also possible.

Annex A

(informative)

Differences from the ISO/IEC 9945 standard

This International Standard originated from the locale and charmap specifications in the ISO/IEC 9945 POSIX, and this International Standard intends to be backwards compatible, so that what is conforming to the POSIX standard should also be valid syntactically and semantically according to the specifications in this International Standard.

A number of enhancements have been made and a number of restrictions have been lifted in comparison to the POSIX standard:

A.1 Restrictions removed

1. Dependence on specific meaning of the character NUL as termination of a string (from the C standard) has been removed, to cater for other programming languages than C.

A.2 Enhancements in the Technical report 2014

1. A description of a "repertoiremap" definition was added to facilitate descriptions of FDCC-sets without charmaps, and also to provide binding from a FDCC-set using one set of character names to charmaps using another naming set.

2. The specific POSIX locale has been replaced with the "i18n" FDCC-set, defined on the repertoire on ISO/IEC 10646.

3. Transliteration support has been added in the LC_CTYPE category.

4. Terminology has been aligned with ISO/IEC TR 11017, especially the POSIX term "locale" has been changed to "FDCC-set".

5. A date escape format "%F" has been added for ISO 8601 dates, and another date escape format "%f" has been added for weekday number with Monday being the first day of the week.

6. Added to LC_MONETARY to accommodate differences between local and international formats:

int_p_cs_precedes
int_p_sep_by_space
int_n_cs_precedes
int_n_sep_by_space

7. Section symbols have been added via the "section-symbol" keyword in the

LC_COLLATE category.

8. The "order_start" keyword has got an optional "section-symbol" identifier

9. The keywords "reorder-section-after" and "reorder-section_end" have been introduced to reorder sections.

10. Symbolic ellipses (both decimal and hexadecimal) has been introduced as a notation.

11. The "print" LC_CTYPE class includes automatically all "graph" characters.

12. The <Uxxxx> and <Uxxxxxxxx> notations have been introduced as predefined symbolic character names, together with a number of symbolic character names derived from POSIX and the Internet.

13. New categories LC_IDENTIFICATION, LC_XLITERATE, LC_NAME, LC_ADDRESS, LC_TELEPHONE, LC_PAPER, LC_MEASUREMENT, and LC_KEYBOARD have been introduced.

14. The LC_CTYPE has got support for new classes, via the new keywords class and map, which corresponds to the C standard library functions iswctype() and towctrans() respectively.

15. The "digit" keyword now supports digits for multiple scripts.

16. The LC_MONETARY category provides support for multiple currencies, such as the native currency and the Euro in some European countries.

17. The LC_TIME has got a number of enhancements to cater for alternate calendars, and timezone information may be given.

18. The charmap specification has been enhanced to support ISO 2022.

A.3 Enhancements in the International Standard

1. New LC_MESSAGES keywords “yesstr” and “nostr”

2. Errors about “country_isbn” and “noexpr” corrected.

3. Note on upgrading data to current versions of ISO/IEC 14651 and Unicode

4. New clauses 7 and 8 on message formats and functionality.

Annex B

(informative)

Rationale

B.1 FDCC-set Rationale

The description of FDCC-sets is based on work performed in the UniForum Technical Committee Subcommittee on Internationalisation and POSIX. Wherever appropriate, keywords were taken from the C Standard or the ISO/IEC 9945 POSIX standard. The C and POSIX term "locale" has been changed into the term "FDCC-set" from ISO/IEC TR 11017 to align with that specification.

The POSIX utility "localedef" compiles locale sources into object files. The "object" definitions need not be portable, as long as "source" definitions are. Strictly speaking, "source" definitions are portable only between applications using the same character set(s). Such "source" definitions can, if they use symbolic names only, easily be ported between systems using different code sets as long as the characters in the portable character set (ISO 646) have common values between the code sets; this is frequently the case in historical applications. Of course, this requires that the symbolic names used for characters outside the portable character set are identical between character sets.

To avoid confusion between an octal constant and a back-reference, the octal, hexadecimal, and decimal constants must contain at least two digits. As single-digit constants are relatively rare, this should not impose any significant hardship. Each of the constants includes "two or more" digits to account for systems in which the byte size is larger than eight bits. For example, an ISO/IEC 10646 system that has defined 16-bit bytes may require six octal, four hexadecimal, and five decimal digits, for some coded characters.

As an international (ISO/IEC) International Standard this International Standard should follow the ISO/IEC guidelines, including the ISO/IEC TR 10176. This TR has a rule that characters outside the invariant part of ISO/IEC 646 should not be used in portable specifications. The backslash and the number-sign character are not in the invariant part. As far as general usage of these symbols, they are covered by the "grandfather clause" specifying previous practice in international standards and in the industry such as in specifications from The Open Group, but for newly defined interfaces, ISO has requested that specifications provide alternate representations, and this International Standard then follows POSIX for backward compatibility. Consequently, while the default escape character remains the backslash, and the default comment character is the number-sign, applications are required to recognize alternative representations, identified in the applicable source text via the "escape_char" and "comment_char" keywords.

B.1.1 LC_IDENTIFICATION Rationale.

The LC_IDENTIFICATION category gives meta-information on the FDCC-set, such as who created it, and what is the level of conformance for each of the FDCC sets.

B.1.2 LC_CTYPE Rationale

The LC_CTYPE category primarily is used to define the encoding-independent aspects of a character set, such as character classification. In addition, certain encoding-dependent characteristics are also defined for an application via the LC_CTYPE category. This International Standard does not mandate that the encoding used in the FDCC-set is the same as the one used by the application, because an application may decide that it is advantageous to define a FDCC-set in a system-wide encoding rather than having multiple, logically identical FDCC-sets in different encodings, and to convert from the application encoding to the system-wide encoding on usage. Other applications could require encoding-dependent FDCC-sets. In either case, the LC_CTYPE attributes that are directly dependent on the encoding, such as "mb_cur_max" and the display width of characters, are not user-specifiable in a locale source, and are consequently not defined as keywords.

As the LC_CTYPE character classes are based on the C Standard character-class definition, the category does not support multi-character elements. For instance, the German character <sharp-s> is traditionally classified as a lowercase letter. There is no corresponding uppercase letter; in proper capitalization of German text the <sharp-s> will be replaced by SS; i.e., by two characters. This kind of conversion is outside the scope of the "toupper" and "tolower" keywords.

The character classes "digit", "xdigit", "lower", "upper", and "space" have a set of automatically included characters. These only need to be specified if the character values (i.e. encoding) differs from the application default values. The definition of character class "digit" allows alternate digits (e.g., Hindi) to be specified here. The definition of character class "xdigit" requires that the characters included in character class "digit" are included here also, and allows for different symbols for the hexadecimal digits 10 through 15.

The "combining" and "combining-level3" classes are an IT-enablement of ISO/IEC 10646 definitions of combining characters. These can be used to check identifiers for consistence with the guidelines given in TR 10176 annex A.

B.1.3 LC_COLLATE Rationale.

The LC_COLLATE category governs the collation order in the FDCC-set, and may thus be useful for the processing of the ISO/IEC 14651 string ordering and comparison standard, the C Standard strxfrm() and strcoll() functions, as well as a number of ISO/IEC 9945 POSIX utilities.

The rules governing collation depends to some extent on the use. At least five different levels of increasingly complex collation rules can be distinguished:

(1) Byte/machine code order. This is the historical collation order in the UNIX system and many proprietary operating systems. Collation is here done character by character, without any regard to context. The primary virtue is that it usually is quite fast, and also completely deterministic; it works well when the native machine collation sequence matches the user expectations.
(2) Character order. On this level, collation is also done character by character, without regard to context. The order between characters is, however, not determined by the code values, but on the user's expectations of the correct order between characters. In addition, such a (simple) collation order can specify that certain characters collate equal (e.g., upper and lowercase letters).
(3) String ordering. On this level, entire strings are compared based on relatively straightforward rules. At this level, several "passes" may be required to determine the order between two strings. Characters may be ignored in some passes, but not in others; the strings may be compared in different directions; and simple string substitutions may be made before strings are compared. This level is best described as "dictionary" ordering; it is based on the spelling, not the pronunciation, or meaning, of the words.
(4) Text search ordering. This is a further refinement of the previous level, best described as "telephone book ordering"; some common homonyms (words spelled differently but with same pronunciation) are collated together; numbers are collated as if spelled with words, and so on.
(5) Semantic level ordering. Words and strings are collated based on their meaning; entire words (such as "the") are eliminated, the ordering is not deterministic. This may requires special software, and is highly dependent on the intended use.

While the historical collation order formally is at level 1, for the English language it corresponds roughly to elements at level 2. The user expects to see the output from the "ls" utility sorted very much as it would be in a dictionary. While telephone book ordering would be an optimal goal for standard collation, this was ruled out as the order would be language dependent. Furthermore, a requirement was that the order must be determined solely from the text string and the collation rules; no external information (e.g., "pronunciation dictionaries") could be required.

As a result, the goal for the collation support is at level 3. This also matches the requirements for the Canadian collation order standard, as well as other, known collation requirements for alphabetic scripts. It specifically rules out collation based on pronunciation rules, or based on semantic analysis of the text. The syntax for the LC_COLLATE category source is the result of a cooperative effort between representatives for many countries and organizations working with international issues, such as UniForum, The Open Group, The Unicode Consortium Inc. and ISO, and it meets the requirements for level 3, and has

been verified to produce the correct result with examples based on Canadian and Danish collation order.

The directives that can be specified in an operand to the order_start keyword are based on the requirements specified in several proposed standards and in customary use. The following is a rephrasing of rules defined for "lexical ordering in English and French" by the Canadian Standards Association (text is brackets is rephrased):

(1) Once special characters (punctuation) have been removed from original strings, the ordering is determined by scanning forward (left to right) [disregarding case and diacriticals].
(2) In case of equivalence, special characters are once again removed from original strings and the ordering is determined scanning backward (starting from the rightmost character of the string and back), character by character, (disregarding case but considering diacriticals).
(3) In case of repeated equivalence, special characters are removed again from original strings and the ordering is determined scanning forward, character by character, (considering both case and diacriticals).
(4) If there is still an ordering equivalence after rules (1) through (3) have been applied, then only special characters and the position they occupy in the string are considered to determine ordering. The string that has a special character in the lowest position comes first. If two strings have a special character in the same position, the character [with the lowest collation value] comes first. In case of equality, the other special characters are considered until there is a difference or all special characters have been exhausted.

It is estimated that the International Standard covers the mechanisms to specify data to cover the requirements for all European languages, and Cyrillic and Middle Eastern scripts.

The Far East (particularly Japanese/Chinese) collations are often based on contextual information. In Japan, collations of strings containing CJK characters (ideograms) are often done considering some related information such as pronunciation, which needs a bulk dictionary (and some common sense). Such collation, in general, falls outside the desired goal of this International Standard, and this International Standard can support only a restricted of collations used in Japan. There are, however, several other collation rules (stroke/radical, or "most common pronunciation") which can be supported with the mechanism described here. Previous drafts contained a substitute statement, which performed a regular expression style replacement before string compares. It has been withdrawn based on balloter objections that it was not required for the types of ordering this International Standard is aimed at.

The character (and collating element) order is defined by the order in which characters and elements are specified between the order_start and order_end keywords. This character order is used in range expressions in regular

expressions. Weights assigned to the characters and elements define the collation sequence; in the absence of weights, the character order is also the collation sequence.

The position keyword was introduced to provide the capability to consider, in a compare, the relative position of non-IGNOREd characters. As an example, consider the two strings "o-ring" and "or-ing". Assuming the hyphen is IGNOREd on the first pass, the two strings will compare equal, and the position of the hyphen is immaterial. On second pass, all characters except the hyphen are IGNOREd, and in the normal case the two strings would again compare equal. By taking position into account, the first collates before the second.

This International Standard adds a number of facilities over the ISO/IEC 9945 POSIX standard, especially in the support for the ISO/IEC 10646 UCS character set. These extended facilities are in alignment with the ISO/IEC 14651 sorting standard. In addition to the facilities provided in ISO/IEC 14651, this specification contains mechanisms to put data into a FDCC-set environment, and has added facilities to sort sections differently, has facilities to reuse FDCC-sets in different notations via the "equivalence-symbol" keyword and tables.

B.1.3.1 "reorder-after" rationale

Much work has been done on FDCC-sets, making them quite general. The ISO/IEC 9945-2:1993 POSIX standard introduced a "copy" command for all categories of the POSIX locale. This is useful for many purposes and it ensures that two FDCC-sets are equivalent for this category. A further step in building on previous FDCC-set work is defined in this International Standard.

Collating sequences often vary a bit from country to country, and from language to language, but generally much of the collating sequence is the same. For example the Danish sequence is for the most part the same as the German or English collation, but for about a dozen letters it differs. The same can be said for Swedish or Hungarian: generally the Latin collating sequence is the same, but a few characters are different.

This International Standard defines a FDCC-set defined on the character repertoire of the ISO/IEC 10646 standard, in a character set independent way. The intention is that some of the information from this FDCC-set will be acceptable in many cultures, and that it can serve as the basis for modifications in other cultures, to obtain a culturally acceptable specification. Using the "reorder-after" construct will also help improve the overview of what the changes really are for implementers and other users.

An example of the use of the "reorder-after" construct is the following. A default international ordering for the Latin alphabet may be adequate for Danish, with the exception of the collation rules for the letters Ü, ü, Æ, æ, Ä, ä, Ø, ø, Ö, ö, Å and å. By applying the "reorder-after" construct, the Danish specification can be made more easily by copying and reordering the existing international

specification, rather than specifying collation parameters for all Latin letters (with or without diacritics). There is no obligation for Denmark to take this approach, but the "reorder-after" construct provides the mechanism for doing so if it is deemed desirable.

B.1.3.2 awk script for "reorder-after" construct

A script has been written in the "awk" language defined in the POSIX standard ISO/IEC 9945 to implement the "reorder-after" construct. It functions as follows: It reads all of the FDCC-set and if in the LC_COLLATE category, it processes the line, else it just outputs the line. For the LC_COLLATE category it reads the lines and puts it into a double linked list of strings identified by a line number; at the end of the LC_COLLATE category all the lines are output. If the line is a "copy" keyword and it reads the file referenced, extracting the LC_COLLATE section of the file in to the list of strings. If the line is a "reorder-after" keyword, it sets a pointer to be the line number of the symbol to of the "reorder-after" keyword. If the line is part of the "reorder-after" specification, it is entered into the double linked list at this point, and the previous entry in the double linked list for the <collation-element> is removed from the list. A "reorder-end" keyword terminates the reordering.

```
BEGIN { comment = "%"; back[0]= follow[0] = 0; }
/LC_COLLATE/ { coll=1 }
/END LC_COLLATE/ { coll=0; for (lnr= 1; lnr; lnr= follow[lnr]) print cont[lnr] }

{ if (coll == 0) print $0  ;
     else { if ($1 == "copy")   {
          file = $2
          while (getline < file )
          if ( $1 == "LC_COLLATE" ) copy_lc = 1
          else if ( $1 == "END" && $2 == "LC_COLLATE" ) copy_lc =0
          else if (copy_lc) {
               lnr++
               follow[lnr-1] = lnr; back [ lnr ] = lnr-1
               cont[lnr] = $0; symb[ $1 ] = lnr
          }
          close (file )
     }
     else if ($1 == "reorder-after") { ra=1 ; after = symb [ $2 ] }
     else if ($1 == "reorder-end") ra = 0
     else {
          lnr++
          if (ra) follow [ lnr ] = follow [ after ]
          if (ra) back [ follow [ after ] ] = lnr
          follow[after] = lnr; back [ lnr ] = after
          cont[lnr] = $0
          if ( ra && $1 != comment && $1 != "" )  {
               old = symb [ $1 ];
               follow [ back [ old ] ] = follow [ old ];
               back [ follow [ old ] ] = back [ old ];
               symb[ $1 ] = lnr;
          }
          after = lnr
     }
     }
}
```

B.1.3.3 Sample FDCC-set specification for Danish

```
escape_char /
comment_char %
repertoiremap "i18nrep"
charset "ISO_8859-1:1987"
% Distribution and use is free, also
% for commercial purposes.

LC_VERSION
title         "Danish language FDCC-set for Denmark"
source        "Danish Standards Association"
address       "Kollegievej 6, DK-2920 Charlottenlund, Danmark"
contact       "Keld Simonsen"
email         "Keld.Simonsen@rap.dk"
tel           "+45 - 3996-6101"
fax           "+45 - 3996-6202"
language      "da"
territory     "DK"
revision      "4.3"
date          "2017-04-05"

category      i18n:2000;LC_IDENTIFICATION
category      i18n:2000;LC_CTYPE
category      i18n:2000;LC_COLLATE
category      i18n:2000;LC_TIME
category      posix:1993;LC_NUMERIC
category      i18n:2000;LC_MONETARY
category      posix:1993;LC_MESSAGES
category      i18n:2000;LC_XLITERATE
category      i18n:2000;LC_NAME
category      i18n:2000;LC_ADDRESS
category      i18n:2000;LC_TELEPHONE

END LC_VERSION

LC_CTYPE
copy "i18n"
END LC_CTYPE

LC_COLLATE
% The ordering algorithm is in accordance
% with Danish Standard DS 377 (1980)
% and the Danish Orthography Dictionary
% (Retskrivningsordbogen, 2. udgave, 1996).
% It is also in accordance with
% Greenlandic orthography.

collating-element <A-A> from "<A><A>"
collating-element <A-a> from "<A><a>"
collating-element <a-A> from "<a><A>"
collating-element <a-a> from "<a><a>"
collating-symbol <SPECIAL>
copy i18n
reorder-after <CAPITAL>
<CAPITAL>
<CAPITAL-SMALL>
<SMALL-CAPITAL>
<SMALL>
reorder-after <q8>
```



```
<kk>     <Q>;<SPECIAL>;<SMALL>;IGNORE
reorder-after <t8>
<TH>     "<T><H>";"<TH><TH>";"<CAPITAL><CAPITAL>";IGNORE
<th>     "<T><H>";"<TH><TH>";"<SMALL><SMALL>";IGNORE
reorder-after <y8>
% <U:> and <U"> are treated as <Y> in Danish
<U:>     <Y>;<U:>;<CAPITAL>;IGNORE
<u:>     <Y>;<U:>;<SMALL>;IGNORE
<U">     <Y>;<U">;<CAPITAL>;IGNORE
<u">     <Y>;<U">;<SMALL>;IGNORE
reorder-after <z8>
% <AE> is a separate letter in Danish
<AE>     <AE>;<NONE>;<CAPITAL>;IGNORE
<ae>     <AE>;<NONE>;<SMALL>;IGNORE
<AE'>    <AE>;<ACUTE>;<CAPITAL>;IGNORE
<ae'>    <AE>;<ACUTE>;<SMALL>;IGNORE
<A3>     <AE>;<MACRON>;<CAPITAL>;IGNORE
<a3>     <AE>;<MACRON>;<SMALL>;IGNORE
<A:>     <AE>;<SPECIAL>;<CAPITAL>;IGNORE
<a:>     <AE>;<SPECIAL>;<SMALL>;IGNORE
% <O//> is a separate letter in Danish
<O//>    <O//>;<NONE>;<CAPITAL>;IGNORE
<o//>    <O//>;<NONE>;<SMALL>;IGNORE
<O//'>   <O//>;<ACUTE>;<CAPITAL>;IGNORE
<o//'>   <O//>;<ACUTE>;<SMALL>;IGNORE
<O:>     <O//>;<DIAERESIS>;<CAPITAL>;IGNORE
<o:>     <O//>;<DIAERESIS>;<SMALL>;IGNORE
<O">     <O//>;<DOUBLE-ACUTE>;<CAPITAL>;IGNORE
<o">     <O//>;<DOUBLE-ACUTE>;<SMALL>;IGNORE
% <AA> is a separate letter in Danish
<AA>     <AA>;<NONE>;<CAPITAL>;IGNORE
<aa>     <AA>;<NONE>;<SMALL>;IGNORE
<A-A>    <AA>;<A-A>;<CAPITAL>;IGNORE
<A-a>    <AA>;<A-A>;<CAPITAL-SMALL>;IGNORE
<a-A>    <AA>;<A-A>;<SMALL-CAPITAL>;IGNORE
<a-a>    <AA>;<A-A>;<SMALL>;IGNORE
<AA'>    <AA>;<AA'>;<CAPITAL>;IGNORE
<aa'>    <AA>;<AA'>;<SMALL>;IGNORE
reorder-end
END LC_COLLATE

LC_MONETARY
int_curr_symbol         "<D><K><K><SP>"
currency_symbol         "<k><r><.>"
mon_decimal_point       "<,>"
mon_thousands_sep       "<.>"
mon_grouping            3;3
positive_sign           ""
negative_sign           "<->"
int_frac_digits         2
frac_digits             2
p_cs_precedes           1
p_sep_by_space          2
n_cs_precedes           1
n_sep_by_space          2
p_sign_posn             4
n_sign_posn             4
END LC_MONETARY
```



```
LC_NUMERIC
decimal_point           "<,>"
thousands_sep           "<.>"
grouping                3;3
END LC_NUMERIC

LC_TIME
abday       "<m><a><n>";/
           "<t><i><r>";"<o><n><s>";/
           "<t><o><r>";"<f><r><e>";/
           "<l><o//><r>";"<s><o/><n>
day         "<m><a><n><d><a><g>";/
           "<t><i><r><s><d><a><g>";/
           "<o><n><s><d><a><g>";/
           "<t><o><r><s><d><a><g>";/
           "<f><r><e><d><a><g>";/
           "<l><o//><r><d><a><g>"/
           "<s><o//><n><d><a><g>";
week        7;19971201;4
abmon       "<j><a><n>";"<f><e><b>";/
           "<m><a><r>";"<a><p><r>";/
           "<m><a><j>";"<j><u><n>";/
           "<j><u><l>";"<a><u><g>";/
           "<s><e><p>";"<o><k><t>";/
           "<n><o><v>";"<d><e><c>"
mon         "<j><a><n><u><a><r>";/
           "<f><e><b><r><u><a><r>";/
           "<m><a><r><t><s>";/
           "<a><p><r><i><l>";/
           "<m><a><j>";/
           "<j><u><n><i>";/
           "<j><u><l><i>";/
           "<a><u><g><u><s><t>";/
           "<s><e><p><t><e><m><b><e><r>";/
           "<o><k><t><o><b><e><r>";/
           "<n><o><v><e><m><b><e><r>";/
           "<d><e><c><e><m><b><e><r>"
d_t_fmt     "<%><a><SP><%><F><SP><%><T><SP><%><Z>"
d_fmt       "<%><O><d><.><SP><%><B><SP><%><Y>"
alt_digits  "<0><.>;<1><.>;<2><.>;<3><.>;<4><.>;/
           <5><.>;<6><.>;<7><.>;<8><.>;<9><.>;/
           <1><0><.>;<1><1><.>;<1><2><.>;<1><3><.>;<1><4><.>;/
           <1><5><.>;<1><6><.>;<1><7><.>;<1><8><.>;<1><9><.>;/
           <2><0><.>;<2><1><.>;<2><2><.>;<2><3><.>;<2><4><.>;/
           <2><5><.>;<2><6><.>;<2><7><.>;<2><8><.>;<2><9><.>;/
           <3><0><.>;<3><1><.>"
t_fmt       "<%><T>"
am_pm       "";""
t_fmt_ampm  ""
timezone    "<C><E><T><-><1><C><E><T><SP><D><S><T><,><M><3><.><5><.><0>/
           <,><M><1><0><.><5><.><0>"
END LC_TIME

LC_MESSAGES
yesexpr     "<<(><1><J><j><Y><y><)/>><.><*>"
noexpr      "<<(><0><N><n><)/>><.><*>"
END LC_MESSAGES

LC_NAME
```



```
name_fmt    "<%><p><%><t><%><g><%><t><%><m><%><t><%><f>"
name_gen    ""
name_mr     "<h><r>"
name_mrs    "<f><r><u>"
name_miss   "<f><r><o/><k><e><n>"
name_ms     "<f><r>"
END LC_NAME

LC_ADDRESS
country_name        "<D><a><n><m><a><r><k>"
country_post        "<D><K>"
lang_ab             "<d><a>"
lang_term           "<d><a><n>"
postal_fmt   "<%><a><%><N><%><f><%><N><%><d><%><N><%><b><%><N><%>/
            <%><s><SP><%><h><SP><%><e><SP><%><r><%><N>/
            <%><C><-><%><z><SP><%><T><%><N><%><c><%><N>"
END LC_ADDRESS

LC_TELEPHONE
tel_int_fmt    "<+><%><c><SP><%><a><SP><%><l>"
tel_dom_fmt    "<%><l>"
int_select     "<0><0>"
int_prefix     "<4><5>"
END LC_TELEPHONE
```

B.1.4 LC_MONETARY Rationale.

The currency symbol does not appear in LC_MONETARY because it is not defined in the C Standard's C locale. The C Standard limits the size of decimal points and thousands delimiters to single-byte values. In FDCC-sets based on multibyte coded character sets this cannot be enforced, obviously; this International Standard does not prohibit such characters, but makes the behaviour unspecified (in the text "In contexts where other standards . . . ").

The grouping specification is based on, but not identical to, the C Standard . The "-1" signals that no further grouping is performed, the equivalent of (CHAR_MAX) in the C Standard).

The FDCC-set definition is an extension of the C Standard localeconv() specification. In particular, rules on how currency_symbol is treated are extended to also cover int_curr_symbol, and p_set_by_space and n_sep_by_space have been augmented with the value 2, which places a space between the sign and the symbol (if they are adjacent; otherwise it should be treated as a 0). The following table shows the result of various combinations:

		p_sep_by_space		
		2	1	0
p_cs_precedes = 1	p_sign_posn = 0	($ 1.25)	($ 1.25)	($1.25)
	p_sign_posn = 1	+ $1.25	+$ 1.25	+$1.25
	p_sign_posn = 2	$1.25 +	$ 1.25+	$1.25+

	p_sign_posn = 3	+ $1.25	+$ 1.25	+$1.25
	p_sign_posn = 4	$ +1.25	$+ 1.25	$+1.25
p_cs_precedes = 0	p_sign_posn = 0	(1.25 $)	(1.25 $)	(1.25$)
	p_sign_posn = 1	+1.25 $	+1.25 $	+1.25$
	p_sign_posn = 2	1.25$ +	1.25 $+	1.25$+
	p_sign_posn = 3	1.25+ $	1.25 +$	1.25+$
	p_sign_posn = 4	1.25$ +	1.25 $+	1.25$+

The following is an example of the interpretation of the mon_grouping keyword. Assuming that the value to be formatted is 123456789 and the mon_thousands_sep is "'", then the following table shows the result. The third column shows the equivalent C Standard string that would be used to accommodate this grouping. It is the responsibility of the utility to perform mappings of the formats in this clause to those used by language bindings such as the C Standard .

Mon_grouping	Formatted Value	C String
3;-1	123456'789	"\3\177"
3	123'456'789	"\3"
3;2;-1	1234'56'789	"\3\2\177"
3;2	12'34'56'789	"\3\2"
-1	123456789	"177"

In these examples, the octal value of (CHAR_MAX) is 177.

The multiple currency support is specified such that a FDCC-set can be used without change during the transition period in a static environment. The specifications can be used without change of the FDCC-set on an installation, when converting from one national currency to another, for example when removing some zeroes to form a new currency.

The following example illustrates the support for multiple currencies; the example is for the Euro in Estonia:

```
LC_MONETARY
valid_from              "";              "20110101"
valid_to                "20101231";      ""
conversion_rate         1/1;             1565/100
int_curr_symbol         "<E><E><K><SP>"; "<E><U><R><SP>"
currency_symbol         "<K><r>";        "<E><U><R>"
mon_decimal_point       "<,>"
mon_thousands_sep       "<.>"
mon_grouping            3;3
positive_sign           ""
negative_sign           "<->"
int_frac_digits         2;               2
frac_digits             2;               2
p_cs_precedes           1;               1
p_sep_by_space          2;               2
```



```
n_cs_precedes           1;              1
n_sep_by_space          2;              2
p_sign_posn             4;              4
n_sign_posn             4;              4

END LC_MONETARY
```

B.1.5 LC_NUMERIC Rationale.

See the rationale for LC_MONETARY (B.1.3) for a description of the behaviour of grouping.

B.1.6 LC_TIME Rationale.

The LC_TIME descriptions of abday, day, and abmon imply a Gregorian style calendar (7-day weeks, 12-month years, leap years, etc.). Other calendars can be supported, for example calendars with a fixed week length.

In some FDCC-sets the field descriptors for weekday and month names will be given with an initial small letter. Programs using these fields may need to adjust the capitalization if the output is going to be used at the beginning of a sentence.

The field descriptors corresponding to the optional keywords consist of a modifier followed by a traditional field descriptor (for instance %Ex). If the optional keywords are not supported by the application or are unspecified for the current FDCC-set, these field descriptors are treated as the traditional field descriptor. For instance, assume the following keywords:

alt_digits "0th";"1st";"2nd";"3rd";"4th";"5th";"6th";"7th";"8th";"9th";"10th"
d_fmt "The %Od day of %B in %Y"

On 1776-07-04, the %x field descriptor would result in "The 4th day of July in 1776," while 1789-07-14 would come out as "The 14 day of July in 1789." It can be noted that the above example is for illustrative purposes only; the %o modifier is primarily intended to provide for Kanji or Hindi digits in date formats. While it is clear that an alternate year format is required, there is no consensus on the format or the requirements. As a result, while these keywords are reserved, the details are left unspecified. It is expected that National Standards Bodies will provide specifications.

B.1.7 LC_MESSAGES Rationale.

The LC_MESSAGES category is described in clause 4 as affecting the language used by utilities for their output. The mechanism used by the application to accomplish this, other than the responses shown here in the FDCC-set definition, is not specified by this version of this International Standard.

B.1.8 LC_XLITERATE Rationale.

Transliteration is often language dependent, transliterating one specific language to another specific language. For example transliteration from Russian to English, and from Serbian to German would normally be quite different, although the same repertoire of characters would be transliterated. Even transliteration of two languages using the same script into one language (for example from Russian to Danish and from Serbian to Danish), or transliteration of the same language (for example Russian into English or German) may be different. The language to be transliterated to is identified with the FDCC-set, which may also be used to identify a specific language to be transliterated from. Transliteration may also be to a specific repertoire of characters, determined for example by limitations of displaying equipment, or what the user can intelligibly read. The capabilities here allows for multiple fallback, so that the specification can be valid for all target character repertoires, eliminating the need for specific data for each target repertoire.

B.1.9 LC_NAME Rationale.

The LC_NAME category gives information to prepare a text for addressing a person, for example as a part of a postal address on an envelope, or as a salutating line in a letter. The information is intended to be given to an API that has the various naming information as parameters and yields a formatted string as the return value.

The "profession" entry is intended for either the general profession of the person in question, or the job title, for use in letters or as part of the address on an envelope.

B.1.10 LC_ADDRESS Rationale.

The LC_ADDRESS category gives information to prepare a text for writing an address, for example as a part of a postal address on an envelope. The information is intended to be given to an API that has the various address information as parameters and yields a formatted string as the return value.

B.1.11 LC_TELEPHONE Rationale.

The LC_TELEPHONE category gives information to prepare a text for writing a telephone number. The information is intended to be given to an API that has the various information on a telephone number as parameters and yields a formatted string as the return value. Both an international and a domestic formatting possibility is available.

B.1.12 LC_PAPER Rationale.

The LC_PAPER category gives information to prepare output on a printer. Only the physical

measurement s of the height and width is available, as this is the information most often available in various document handling applications.

B.1.13 LC_MEASUREMENT Rationale.

The LC_MEASUREMENT category gives a simple indication whether the ISO measurement system is used, or another systems is the one applied. It may be enhanced in future editions of this standard.

B.1.13 LC_KEYBOARD Rationale.

The LC_KEYBOARD category gives a way to identify relevant keyboards for the FDCC-set. Applications may chose to use the first keyboard specified, if no explicit keyboard identification is given, or suggest the first keyboard specified if a keyboard selection is to be carried out.

B.2 Character Set Rationale.

This International Standard poses no requirement that multiple character sets or code sets be supported, leaving this as a marketing differentiation for implementors. Although multiple charmaps are supported, it is the responsibility of the application to provide the file(s); if only one is provided, only that one will be accessible.

The character set description text provides the capability to describe character set attributes (such as collation order or character classes) independent of character set encoding, and using only the characters in the portable character set. This makes it possible to create "generic" FDCC-set source texts for all code sets that share the portable character set (such as the ISO/IEC 8859 family or IBM Extended ASCII).

Applications are free to describe more than one code set in a character set description text. For example, if an application defines ISO/IEC 8859-1 as the primary code set, and ISO/IEC 8859-2 as an alternate set, with each character from the alternate code set preceded in data by a shift code, a character set description text could contain a complete description of the primary set and those characters from the secondary that are not identical, the encoding of the latter including the shift code.

Applications are free to choose their own symbolic names, as long as the names identified by this International Standard are also defined; this provides support for already existing "character names".

The charmap was introduced to resolve problems with the portability of, especially, FDCC-set sources. While the portable character set (in Table 1) is a constant across all FDCC-sets for a particular application, this is not true for the extended character set. However, the particular coded character set used for an application does not necessarily imply different characteristics or collation:

on the contrary, these attributes should in many cases be identical, regardless of codeset. The charmap provides the capability to define a common FDCC-set definition for multiple codesets (the same FDCC-set source can be used for codesets with different extended characters; the ability in the charmap to define ``empty'' names allows for characters missing in certain codesets).

In addition, some implementers have expressed an interest in using the charmap to define certain other characteristics of codesets, such as the <mb_cur_max> value for the particular codeset. (Note that <mb_cur_max> has to be equal to or lower than the C Standard {MB_LEN_MAX}, which is the application limit). Such extensions are not described here; but may be added in a later revision of this International Standard.

The <escape_char> declaration was added at the request of the international community to ease the creation of portable charmaps on terminals not implementing the default backslash escape. (This approach was adopted because this was a new interface invented by ISO/IEC 9945-2:1993 POSIX. Historical interfaces, such as the shell command language and awk, have not been modified to accommodate this type of terminal.)

The octal number notation was selected to match those of POSIX "awk" and "tr" utilities and is consistent with that used by the POSIX localedef utility.

The charmap capability implements a facility available at some X/Open compatible applications. Its prime virtue is to support "generic" collation sequence source definitions. An implementer or an applications developer can produce a template definition that can be used to produce several codeset-dependent "compiled" FDCC-set definitions. The facility also removes any dependency in many source definitions on characters outside the character set defined in this clause.

The charmap allows specification of more than one encoding of a character. This allows for encodings that can encode items in more than one way. For example, an item can be encoded once as a fully composed character and again as a base character plus combining character. This would allow either representation to be recognized. As only the first occurrence of the character may be output, this technique could be used to normalize a character stream.

The ISO 2022 support introduced gives the possibility to refer other definitions via charmaps, so the full encoding does not have to be replicated. It supports shifting with G0, G1, G2 and G3 sets, and also general shifting of coded character sets via escape sequences.

B.3 Repertoiremap Rationale.

The repertoiremap was introduced to make FDCC-sets independent of the availability of charmaps. With the repertoiremap it is possible to use a FDCC-set

encoded with one set of symbolic character names, together with charmaps with other symbolic character naming schemes, provided there are repertoiremaps available for both naming schemes.

Repertoiremaps are also useful to describe repertoires of characters, to be used for example for transliteration, printing or display.

Annex C
(informative)

BNF Grammar

C.1 BNF Syntax Rules

The syntax used here is near to ISO/IEC 14977, but "_" is allowed in identifiers, and comma is not used as concatenator, as the items are just concatenated.

Definitions between <angle brackets> make use of terms not defined in this BNF syntax, and assume general English usage.

Other conventions:

* means 0 or more repetitions of a token.
+ means one or more repetitions of a token
Brackets [ ] indicate optional occurrence of a token.
Comments start with a % on a separate line.

There may be more specifications in the normative text that describes restrictions on the grammar.

C.2 Grammar for FDCC-sets

```
% The following is the overall FDCC-set grammar
FDCC_set_definition         = [ global_statement* ] category+ ;
global_statement            = 'escape_char' SP char_symbol EOL
                            | 'comment_char' SP char_symbol end_of_line
                            | 'repertoiremap' SP quoted_string EOL
                            | 'charmap' SP quoted_string EOL ;
category                    = lc_identification | lc_ctype | lc_collate
                            | lc_monetary | lc_numeric | lc_time
                            | lc_messages | lc_xliterate | lc_telephone
                            | lc_name | lc_address | lc_paper
                            | lc_measurement | lc_keyboard ;

% The following is the LC_IDENTIFICATION category grammar
lc_identification           = ident_head ident_keyword* ident_tail
                            | ident_head copy_FDCC_set ident_tail ;
ident_head                  = 'LC_IDENTIFICATION' EOL ;
ident_keyword               = ident_keyword_string SP quoted_string EOL ;
ident_keyword_string        = 'title' | 'source' | 'address' | 'contact'
                            | 'email' | 'tel' | 'fax'| 'language' | 'script'
                            | 'territory' | 'audience' | 'application'
                            | 'abbreviation' | 'revision' | 'date' ;
ident_tail                  = 'END' SP 'LC_IDENTIFICATION' EOL ;

% The following is the LC_CTYPE category grammar
lc_ctype                    = ctype_head ctype_keyword* ctype_tail
                            | ctype_head copy_FDCC_set ctype_tail ;
ctype_head                  = 'LC_CTYPE' EOL ;
ctype_keyword               = charclass_keyword SP charclass_list EOL
```



```
                        | charconv_keyword SP charconv_list EOL
                        | 'width' SP width_list EOL
                        | translit_section ;
charclass_keyword       = 'upper' | 'lower' | 'alpha' | 'digit'
                        | 'alnum' | 'punct' | 'xdigit' | 'space' | 'print'
                        | 'graph' | 'blank' | 'cntrl' | 'outdigit'
                        | 'class' charclass_name semicolon ;
charclass_name          = '"combining"' | '"combining_level3"'
                        | '"' identifier '"' ;
charclass_list          = charclass_list semicolon char_symbol
                        | charclass_list semicolon ctype_abs_ellipsis
                        semicolon char_symbol
                        | charclass_list semicolon charsymbol
                        ctype_symbolic_ellipses charsymbol
                        | char_symbol ;
width_list              = charclass_list ':' number
                        | width_list semicolon width_list ;
charconv_keyword        = 'toupper' | 'tolower'
                        | 'map' '"' identifier '"' semicolon ;
charconv_list           = charconv_list semicolon charconv_entry
                        | charconv_entry ;
charconv_entry          = '(' char_symbol comma char_symbol ')' ;
translit_section        = 'translit_start' EOL translit_body  'translit_end'
EOL ;
ctype_symbolic_ellipses = '..' | '....' | '..(2)..';
ctype_abs_ellipses      = '...' ;
ctype_tail              = 'END' SP 'LC_TYPE' EOL ;

% The following is the LC_COLLATE category grammar
lc_collate              = collate_head collate_keywords collate_tail ;
collate_head            = 'LC_COLLATE' EOL ;
collate_keywords        = opt_statement* order_statements | delta ;
opt_statement           = 'collating-symbol' SP collsymbol_list EOL
                        | 'collating-element' SP collelement SP 'from' SP
                        collelem_string EOL
                        | 'section-symbol' space+ section_symbol EOL
                        | 'col_weight_max' SP number EOL
                        | 'symbol-equivalence' SP collsymbol SP collsymbol
                        EOL
                        | collation_statement ;
collelem_string         = '"' char_symbol+ '"';
order_statements        = order_start collation_order order_end ;
order_start             = 'order_start' SP order_params EOL ;
order_params            = [section_symbol] [semicolon order_opts] ;
order_opts              = order_opt [ semicolon order_opt ]* ;
order_opt               = opt_word [ comma opt_word ]* ;
opt_word                = 'forward' | 'backward' | 'position' ;
section                 = 'section' SP section_symbol [ SP collsymbol_list ]
EOL ;
collation_order         = ( order_start | section | collation_statement)* ;
collation_statement     = collsymbol EOL
                        | collating_element [ SP weight_list ] EOL ;
collsymbol_list         = collsymbol_element
                        [ semicolon collsymbol_element ]* ;
collsymbol_element      = collsymbol
                        | collsymbol SP ellipses SP collsymbol ;
collating_element       = char_symbol | collelement
                        | ellipses | 'UNDEFINED' ;
weight_list             = weight_symbol [ semicolon weight_symbol ]* ;
```



```
weight_symbol               = <empty>
                            | char_symbol
                            | collsymbol
                            | '"' elem_list '"'
                            | '"' symb_list '"'  | 'IGNORE' ;
ellipses                    = '...'  | '..' | '....' ;
order_end                   = 'order_end' EOL ;
delta                       = opt_statement*
                              'copy' SP FDCC_set_name EOL
                             opt_statement*
                             reordering_statement* ;
reordering_statement        = reorder_after_block
                            | reorder_section_after_1
                            | reorder_section_block ;
reorder_after_block         = reorder_after (collation_order | reorder_after)*
reorder_end ;
reorder_after               = 'reorder-after' SP collsymbol EOL ;
reorder_end                 = 'reorder-end' EOL ;
reorder_section_block       = reorder_section_after_2 section_statement*
reorder_section_end ;
section_statement           = section_symbol SP order_opts EOL ;
reorder_section_after_1     = 'reorder-section-after' SP sectionsymbol
                            SP collsymbol EOL;
reorder_section_after_2     = 'reorder-section-after' SP collsymbol EOL;
reorder_section_end         = 'reorder-section-end' EOL ;
collate_tail                = 'END' SP 'LC_COLLATE' EOL

% The following is the LC_MESSAGES category grammar
lc_messages                 = messages_head messages_keyword* messages_tail
                            | messages_head copy_FDCC_set messages_tail ;
messages_head               = 'LC_MESSAGES' EOL ;
messages_keyword            = 'yesexpr' SP '"' extended_reg_expr '"' EOL
                            | 'noexpr' SP '"' extended_reg_expr '"' EOL
                            | 'yesstr' SP quoted-string EOL
                            | 'nostr' SP quoted-string EOL
                            ;
messages_tail               = 'END' SP 'LC_MESSAGES' EOL ;

% The following is the LC_MONETARY category grammar
lc_monetary                 = monetary_head monetary_keyword* monetary_tail   |
monetary_head copy_FDCC_set monetary_tail ;
monetary_head               = 'LC_MONETARY' EOL ;
monetary_keyword            = mon_keyword_string SP quoted_string EOL
                            | mon_keyword_strings SP mon_string_list EOL
                            | mon_keyword_char SP mon_number_list EOL
                            | mon_keyword_date SP mon_date_list EOL
                            | 'conversion_rate' SP mon_conv_list EOL
                            | 'mon_grouping' SP mon_group_list EOL ;
mon_keyword_string          = 'mon_decimal_point' | 'mon_thousands_sep'
                            | 'positive_sign' | 'negative_sign' ;
mon_keyword_strings         = 'int_curr_symbol' | 'currency_symbol' ;
mon_keyword_char            = 'int_frac_digits' | 'frac_digits'
                            | 'p_cs_precedes' | 'p_sep_by_space'
                            | 'n_cs_precedes' | 'n_sep_by_space'
                            | 'int_p_cs_precedes' | 'int_p_sep_by_space'
                            | 'int_n_cs_precedes' | 'int_n_sep_by_space'
                            | 'p_sign_posn' | 'n_sign_posn'
                            | 'int_p_sign_posn' | 'int_n_sign_posn' ;
mon_keyword_date            = 'valid_from' | 'valid_to' ;
```



```
mon_date_list               = mon_date | mon_date_list semicolon mon_date ;
mon_date                    = '"' 8 * digit '"' ;
mon_group_list              = number   | mon_group_list semicolon number ;
mon_string_list             = quoted_string [ semicolon quoted_string]* ;
mon_number_list             = mon_number | mon_number_list semicolon
mon_number ;
mon_number                  = number | -1 ;
mon_conv_list               = mon_pair | mon_conv_list semicolon mon_pair ;
mon_pair                    = number spaces* '/' spaces* number ;
monetary_tail               = 'END' SP 'LC_MONETARY' EOL ;

% The following is the LC_NUMERIC category grammar
lc_numeric                  = numeric_head numeric_keyword* numeric_tail
                            | numeric_head copy_FDCC_set numeric_tail ;
numeric_head                = 'LC_NUMERIC' EOL ;
numeric_keyword             = num_keyword_string SP quoted_string EOL
                            | num_keyword_grouping SP num_group_list EOL ;
num_keyword_string          = 'decimal_point' | 'thousands_sep' ;
num_keyword_grouping        = 'grouping' ;
num_group_list              = number
                            | num_group_list semicolon number ;
numeric_tail                = 'END' SP 'LC_NUMERIC' EOL ;

% The following is the LC_TIME category grammar
lc_time                     = time_head time_keyword* time_tail
                            | time_head copy_FDCC_set time_tail ;
time_head                   = 'LC_TIME' EOL ;
time_keyword                = time_keyword_name SP time_list EOL
                            | time_keyword_fmt SP quoted_string EOL
                            | time_keyword_opt SP time_list EOL
                            | 'week' SP number semicolon mon_date semicolon
                            number EOL
                            | time_keyword_num SP number EOL
                            | 'timezone' SP time_list EOL;
time_keyword_name           = 'abday' | 'day' | 'abmon' | 'mon' | 'am_pm' ;
time_keyword_fmt            = 'd_t_fmt' | 'd_fmt' |'t_fmt' | 't_fmt_ampm';
time_keyword_opt            = 'era' |'era_year'| 'era_d_fmt'| 'alt_digits' |
era_d_t_fmt | era_t_fmt ;
time_keyword_week           = 'week' ;
time_keyword_num            = 'first_weekday' | 'first_workday'
                            | 'cal_direction' ;
time_list                   = time_list semicolon quoted_string
                            | quoted_string ;
time_tail                   = 'END' SP 'LC_TIME' EOL ;

% The following is the LC_XLITERATE category grammar
lc_xliterate                = translit_head translit_body translit_tail ;
translit_body               = [translit_include] [default_missing]
                            translit_statement* | copy_FDCC_set ;
translit_head               = 'LC_XLITERATE' EOL ;
translit_include            = 'include' SP FDCC_set_name
                            semicolon quoted_nonempty_string EOL ;
default_missing             = 'default_missing' SP quoted_string EOL ;
translit_ignore             = 'translit_ignore' SP charclass_list EOL ;
translit_statement          = char_or_string SP char_or_string
                            [ semicolon char_or_string ]*  EOL ;
translit_tail               = 'END' SP 'LC_XLITERATE' EOL ;

% The following is the LC_NAME category grammar
```



```
lc_name                     = name_head name_keyword* name_tail
                            | name_head copy_FDCC_set name_tail ;
name_head                   = 'LC_NAME' EOL ;
name_keyword                = name_keyword_string SP quoted_string EOL ;
name_keyword_string         = 'name_fmt' | 'name_gen' | 'name_mr'
                            | 'name_mrs' | 'name_ms' | 'name_miss'
                            | 'name_ms' ;
name_tail                   = 'END' SP 'LC_NAME' EOL ;

% The following is the LC_ADDRESS category grammar
lc_address                  = address_head address_keyword* address_tail
                            | address_head copy_FDCC_set address_tail ;
address_head                = 'LC_ADDRESS' EOL ;
address_keyword             = address_keyword_string SP quoted_string EOL ;
address_keyword_string      = 'postal_fmt' | 'country_name' | 'country_post'
                            |  'country_isbn' |'lang_name' | 'lang_ab2'
                            | 'lang_ab3_term' | 'lang_ab3_lib' ;
address_tail                = 'END' SP 'LC_ADDRESS' EOL ;

% The following is the LC_TELEPHONE category grammar
lc_telephone                = tel_head tel_keyword* tel_tail
                            | tel_head copy_FDCC_set tel_tail ;
tel_head                    = 'LC_TELEPHONE' EOL ;
tel_keyword                 = tel_keyword_string SP quoted_string EOL ;
tel_keyword_string          = 'tel_int_fmt' | 'tel_dom_fmt' | 'int_select'
                            | 'int_prefix' ;
tel_tail                    = 'END' SP 'LC_TELEPHONE' EOL ;

% The following is the LC_PAPER category grammar
lc_paper                    = paper_head paper_keyword* paper_tail
                            | paper_head copy_FDCC_set paper_tail ;
paper_head                  = 'LC_PAPER' EOL ;
paper_keyword               = paper_keyword_num NUMBER EOL ;
paper_keyword_num           = 'height' | 'width' ;
paper_tail                  = 'END' SP 'LC_PAPER' EOL ;

% The following is the LC_MEASUREMENT category grammar
lc_measurement              = measurement_head measurement_keyword*
                            measurement_tail
                            | measurement_head copy_FDCC_set measurement_tail ;
measurement_head            = 'LC_MEASUREMENT' EOL ;
measurement_keyword         = measurement_keyword_num NUMBER EOL ;
measurement_keyword_num     = 'measurement' ;
measurement_tail            = 'END' SP 'LC_MEASUREMENT' EOL ;

% The following is the LC_KEYBOARD category grammar
lc_keyboard                 = keyboard_head keyboard_keyword* keyboard_tail
                            | keyboard_head copy_FDCC_set keyboard_tail ;
address_head                = 'LC_KEYBOARD' EOL ;
keyboard_keyword            = keyboard_keyword_string SP quoted_string EOL ;
keyboard_keyword_string     = 'keyboards' ;
keyboard_tail               = 'END' SP 'LC_KEYBOARD' EOL ;


% The following grammar rules are common to all categories
char                        = <any character except those that makes an
                            End Of Line>
graphic_char                = <any char except control_chars and space> ;
space                       = ' ' | <TAB> ;
```



```
SP                          = space+ ;
EOL                         = end_of_line | comment end_of_line ;
end_of_line                 = <anything that makes an End Of Line (EOL)
                            in the operating system employed> ;
comment_char                = <defined by the 'comment_char' keyword> ;
escape_char                 = <defined by the 'escape_char' keyword> ;
charsymbol                  = simple_symbol | ucs_symbol ;
collsymbol                  = simple_symbol ;
collelement                 = simple_symbol ;
sectionsymbol               = simple_symbol ;
octdigit                    = '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7' ;
digit                       = '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9' ;
hex_upper                   = 'A'|'B'|'C'|'D'|'E'|'F'| digit ;
hexdigit                    = hex_upper | 'a'|'b'|'c'|'d'|'e'|'f' ;
letter                      = 'a'|'b'|'c'|'d'|'e'|'f'|'g'|'h'|'i'|'j'|'k'
                            |'l'|'m'|'n'|'o'|'p'|'q'|'r'|'s'|'t'|'u'|'v'
                            |'w'|'x'|'y'|'z'|'A'|'B'|'C'|'D'|'E'|'F'|'G'
                            |'H'|'I'|'J'|'K'|'L'|'M'|'N'|'O'|'P'|'Q'|'R'
                            |'S'|'T'|'U'|'V'|'W'|'X'|'Y'|'Z' ;
portable_graph_gtr          = letter | digit | '!'|'"'|'#'|'$'|'%'|'&'
                            | "'"|'('|')'|'*'|'+'|','|'-'|'.'|'/'|':'|';'
                            | '<'|'='|'?'|'@'|'['|'\'|']'|'^'|'_'
                            | '`'|'{'|'|'|'}'|'~' ;
portable_graph              = portable_graph_gtr | '>' ;
portable_char               = portable_graph | ' ' | <NUL> | <ALERT>
                            | <BACKSPACE> | <TAB> | <CARRIAGE_RETURN>
                            | <NEWLINE> | <VERTICAL_TAB> | <FORM_FEED> ;
octal_char                  = escape_char     octdigit octdigit octdigit* ;
hex_char                    = escape_char 'x' hexdigit hexdigit hexdigit* ;
decimal_char                = escape_char 'd' digit digit digit* ;
number                      = digit+ ;
id_part                     = letter | digit | '-' | '_' ;
four_digit_hex_string       = hex_upper hex_upper hex_upper hex_upper ;
identifier                  = letter id_part* ;
simple_symbol               = space* '<' portable_graph_gtr+ '>' ;
ucs_symbol                  = space* '<U' four_digit_hex_string
                            [ four_digit_hex_string ] '>' ;
quoted_string               = '"' char_symbol* '"' ;
quoted_nonempty_string      = '"' char_symbol+ '"' ;
char_symbol                 = char | charsymbol
                            | octal_char | hex_char | decimal_char ;
elem_list                   = elem+ ;
elem                        = char_symbol | collsymbol | collelement ;
symb_list                   = collsymbol+ ;
FDCC_set_name               = FDCC-name | '"' FDCC-name '"' ;
copy_FDCC_set               = 'copy' FDCC_set_name EOL ;
FDCC-name                   = portable_graph+ ;
semicolon                   = space* ';' space* ;
comma                       = space* ',' space* ;
comment                     = comment_char char* ;
```


Annex D

(informative)

Relation to taxonomy

The following lists the relation for items in this specification to the internationalization taxonomy described in ISO/IEC TR 24785:2009.

Code	Title	Clause
/ (no id)	TAXONOMY	
L/	**LOCALES**	**4**
L/1	**Specifications**	
L/1.1	Languages	
L/1.1.1	Natural languages	4.8
L/1.1.1.1	Vocabulary	
L/1.1.1.1.1	Standard terminology	
L/1.1.1.1.2	Thesauri	
L/1.1.1.1.3	Standard phrases	4.8
L/1.1.1.1.4	Translation	
L/1.1.1.2	Grammar	
L/1.1.1.3	Orthography	
L/1.1.1.3.1	Alphabet	4.3
L/1.1.1.3.2	Spelling	
L/1.1.1.3.3	Use of special characters	4.3
L/1.1.1.3.4	Capitalization	4.3
L/1.1.1.3.5	Hyphenation	
L/1.1.1.3.6	Punctuation	4.3
L/1.1.1.3.7	Transcription	4.9
L/1.1.1.3.8	Ordering	4.4
L/1.1.1.3.9	Personal names and titles	4.10
L/1.1.1.4	Speech	
L/1.2	Cultural conventions	all
L/1.2.1	Cultural elements	all
L/1.2.1.1	Orthography	
L/1.2.1.1.1	Date and time format	4.7
L/1.2.1.1.2	Numeric separators	4.6
L/1.2.1.1.3	Monetary format	4.7
L/1.2.1.1.4	Telephone number format	4.12
L/1.2.1.1.5	Payment number format	

L/1.2.1.1.6	Mail address format	4.11
L/1.2.1.1.7	National places	
L/1.2.1.2	Measurement system	4.14
L/1.2.1.3	Layout styles	
L/1.2.1.4	Paper sizes	4.13
L/1.2.1.5	Use of graphical symbols	
L/1.2.1.6	Use of colours	
L/1.2.1.7	Artistic requirements	
L/1.2.1.8	Jurisdictional requirements	
L/1.2.2	Multilingual requirements	
L/1.3	Operating system dependency	
L/1.3.1	POSIX	
L/1.3.2	Other TOG	
L/2	**Registration**	
L/2.1	Procedures	
L/2.1.1	Supra-national	
L/2.1.1.1	National	
L/2.1.2	World-wide	
L/3	**Implementation**	
L/3.1	Fallback	4.9
C/	**CHARACTERS**	**5**
C/1	**Character information**	**5**
C/1.1	Identification	5
C/1.1.1	Characters	5
C/1.1.1.1	Identifiers	4.3
C/1.1.1.2	Attributes	4.3
C/1.1.2	Repertoires	6
C/1.1.2.1	Graphic characters	4.3
C/1.1.2.1.1	Natural language alphabets	
C/1.1.2.1.1.1	Supra-national	
C/1.1.2.1.1.1.1	General	
C/1.1.2.1.1.1.2	Elderly/disabled	
C/1.1.2.1.1.2	World-wide	
C/1.1.2.1.2	Programming language alphabets	4.3
C/1.1.2.1.3	Non-alphabetic symbols	4.3
C/1.1.2.1.3.1	General	
C/1.1.2.1.3.2	Disabled/elderly	
C/1.1.2.2	Control functions	4.3
C/1.1.2.3	Registration	

C/1.1.3	Glyphs	
C/1.1.3.1	Registration	
C/1.1.3.2	Character correspondence	
C/1.1.4	Glyph repertoires	
C/1.1.4.1	Registration	
C/1.1.4.2	Repertoire correspondence	
C/1.2	Manipulation	
C/1.2.1	Transformation	
C/1.2.1.1	Case conversion	
C/1.2.1.2	Transliteration	
C/1.2.1.3	Fallback representation	
C/2	**Input/output**	
C/2.1	Input	
C/2.1.1	Keyboard	4.15
C/2.1.2	Other means	
C/2.2	Output	
C/2.2.1	Character repertoires	6
C/2.2.2	Character attributes	4.3
C/3	**Electronic processing**	
C/3.1	Processing of coding schemes	
C/3.1.1	Encoding of graphic characters	
C/3.1.2	Encoding of control functions	
C/3.1.3	Code transformations	
C/3.1.3.1	UCS--UCS	
C/3.1.3.2	UCS--other coding schemes	
C/3.2	Interchange/communication	
C/3.2.1	7-bit method	
C/3.2.2	8-bit method	
C/3.2.3	Multiple-octet method	
C/3.3	Internationalization support	
C/3.3.1	Programming languages	
C/3.3.1.1	Language-dependent	
C/3.3.1.2	Language-independent	
C/3.3.2	Operating systems	
C/3.3.3	Communications	
C/3.3.3.1	Directory services	
C/3.3.3.2	Telematics	

Annex E

(informative)

Implementation in glibc

The following constructs of the Technical Report from 2014 were not recognized by the localedef compilation program of GNU libc version 2.11:

4.2 LC_IDENTIFICATION keyword “script” is not recognized.

4.5 LC_MONETARY keywords “valid_from”, “valid_to” and “conversion_rate” are not recognized. With keywords “currency_symbol” and “int_curr_symbol”, more than one currency symbol is not recognized.

4.7 LC_TIME keyword “timezone” not fully supported.

4.9 LC_XLITERATE is not supported, but the keywords “translit_start” and “translit_end” of 4.3 are supported, which uses the specifications in 4.9.

4.15 LC_KEYBOARD is not supported.

Annex F

(informative)

Relation between categories and keywords, and APIs

The relations between the 3 clauses: FDCC-sets, charmaps and repertoiremaps, and APIs are given in the following.

Table: Relation between categories and keywords, and APIs

category	keyword	API
LC_ALL	all keywords	intlocaleinfo stringlocaleinfo
LC_CTYPE	upper lower alpha	istype
	digit outdigit space	istype
	cntrl punct graph	istype
	print xdigit blank	istype
	toupper	touppers stringtrans
	tolower	tolowers stringtrans
	class	istype
	map	istype
LC_COLLATE	all	stringcoll stringncoll
LC_MONETARY	all	money2sting
LC_NUMERIC	all	real2string string2real
LC_TIME	all	time2string string2time
LC_MESSAGES	all	stringget
LC_XLITERATE	all	stringtrans
	translit_start translit_end	stringtrans
	include default_missing	stringtrans
LC_TELEPHONE	tel_dom_fmt	teldom2string
	tel_int_fmt	telint2string
LC_IDENTIFICATION	all	stringlocaleinfo

APIs addressing charmaps:
APIs addressing repertoiremaps:

Annex G

(informative)

Bindings guidelines

This annex gives guidelines for binding to other programming languages.

The APIs of this document are written as a specification in C.

Another programming language (PL) binding to the definition in C notation may be done by a reference to the clause number where the procedure is defined, and then the PL API name is bound to the C API, the PL parameters are bound to the C parameters in the same sequence, and the PL return value is bound to the C result value. The C specification clause may then be used as part of the specification of the binding technique. The binding then has the same semantics as the API described in the mentioned C specification clause. The string type and other data types should be bound to the appropriate internal character representation in the PL.

Example of a Pascal binding, given the appropriate definitions of data structures:

The C specification is

5.4.1 int newlocale(int category_mask, const string localename,locale lc)

Then a Pascal binding could be:

5.4.1 integer procedure newlc(integer cat; string lcname; locale lc)

.

Annex H

(informative)

Index

BIBLIOGRAPHY

The following specifications are considered relevant to this International Standard, in addition to the normative references.

ISO/IEC TR 11017:1998, *Information technology -- Framework for internationalization*

ISO/IEC TR 24785:2009, *Information technology -- Taxonomy of cultural and linguistic adaptability user requirements*

The Unicode Consortium: *The Unicode Standard, Version 5.0*, Addison-Wesley Professional, 27 October 2006. ISBN 0-321-48091-0.

The Unicode Consortium: *UTS #35 - Unicode Locale Data Markup Language (LDML)*, http://unicode.org/reports/tr35/

The Unicode Consortium: *CLDR – Unicode Common Locale Data repository*, http://cldr.unicode.org/

IBM: *National Language Design Guide Volume 2 - National Language Support Reference Manual*, IBM SE09-8002-03, August 1994.

GNU Public License, http://www.gnu.org/licenses/gpl-2.0.html 1991

GNU libc (glibc), http://www.gnu.org/software/libc/

CEPT, CEPT-MAILCODE, *Country code for mail*.

STRÍ: *Nordic Cultural Requirements on Information Technology (Summary report)*, STRÍ TS3, Libris, Reykjavík, Iceland 1992. ISBN 9979-9004-3-1.

Gettext, http://www.gnu.org/software/**gettext**/

