Add a step parameter to iota_view

Abstract

We propose adding a step parameter to iota_view.

Proposal

We propose adding a step parameter to iota_view such that

- views::iota can generate increasing values with an increment different than 1
- views::iota can generate decreasing values, by the mean of a negative increment (-1 or arbitrary integral value)

view::iota(0, 10, 2); // [0, 4, 6, 8]
view::iota(10, 0, -2); // [10, 8, 6, 6, 2]
view::iota(0, unreacheable_t{}, -1) | views::take(5); // [0, -1, -2, -3, -4]

Motivation

I suggested including a step parameter to iota_view in Belfast while discussing P1894 [2]. This would allow views::iota to offer a feature set equivalent to that of Python's range function. It is notably useful to iterate over interlaced data such as matrices, images or audio signals. It can also be used to implements counters, paging and any other use case requiring linear sequences. That feature has been requested in Range-v3 [RangeV3].

However, iota_view can be expressed in terms of view::stride [1] and views::reverse. While iota_view can be more efficient (and less verbose) than combining multiple views, a combination of views::stride and views::enumerate should be preferred over using iota_view when iterating over a container.

As such, despite the relative simplicity of the proposal and the general usefulness of the feature demonstrated by other languages and libraries, it is unclear that the cost/benefit
ratio of the present proposal plays in its favor. In any case, `view::stride` [1] or something similar [Conor] should be prioritized over the present proposal.

**Implementation**

An implementation is available [https://github.com/ericniebler/range-v3/pull/1392](https://github.com/ericniebler/range-v3/pull/1392)

**Wording**

No wording is provided at this time, the paper intends to gauge interest.

**References**


[Conor] CppCon 2019: Conor Hoekstra “23 Ranges: slide & stride” [https://www.youtube.com/watch?v=_lqZJK2vjI](https://www.youtube.com/watch?v=_lqZJK2vjI)