Module std::u128

Returns the bitwise not of the value. Each bit that is 1 becomes 0. Each bit that is 0 becomes 1.

Return the larger of x and y

Return the smaller of x and y

Return the absolute value of x - y

Calculate x / y, but round up the result.

Return the value of a base raised to a power

Get a nearest lower integer Square Root for x. Given that this function can only operate with integers, it is impossible to get perfect (or precise) integer square root for some numbers.

Example:

In integer math, one of the possible ways to get results with more precision is to use higher values or temporarily multiply the value by some bigger number. Ideally if this is a square of 10 or 100.

Example:

Try to convert a u128 to a u8 . Returns None if the value is too large.

Try to convert a u128 to a u16 . Returns None if the value is too large.

Try to convert a u128 to a u32 . Returns None if the value is too large.

Try to convert a u128 to a u64 . Returns None if the value is too large.

Maximum value for a u128

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (exclusive)

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (inclusive)

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

Function

Returns the bitwise not of the value. Each bit that is 1 becomes 0. Each bit that is 0 becomes 1.

```bash
```

```bash
```

Return the larger of x and y

```bash
```

```bash
```

Return the smaller of x and y

```bash
```

```bash
```

Return the absolute value of x - y

```bash
```

```bash
```

Calculate x / y, but round up the result.

```bash
```

```bash
```

Return the value of a base raised to a power

```bash
```

```bash
```

Get a nearest lower integer Square Root for x. Given that this function can only operate with integers, it is impossible to get perfect (or precise) integer square root for some numbers.

Example:

In integer math, one of the possible ways to get results with more precision is to use higher values or temporarily multiply the value by some bigger number. Ideally if this is a square of 10 or 100.

Example:

```bash
```

```bash
```

Try to convert a u128 to a u8 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u16 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash
```

```bash
```

```bash
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f to each number from start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash
```

```bash
```

Function

Return the larger of x and y

```bash
```

```bash
```

Return the smaller of x and y

```bash
```

```bash
```

Return the absolute value of x - y

```bash
```

```bash
```

Calculate x / y, but round up the result.

```bash
```

```bash
```

Return the value of a base raised to a power

```bash
```

```bash
```

```

Get a nearest lower integer Square Root for x. Given that this function can only operate with integers, it is impossible to get perfect (or precise) integer square root for some numbers.

Example:

In integer math, one of the possible ways to get results with more precision is to use higher values or temporarily multiply the value by some bigger number. Ideally if this is a square of 10 or 100.

Example:

```bash

```

```bash

```

Try to convert a u128 to a u8 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u16 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash

```

```bash

```

```bash

```

```bash

```
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (inclusive)

```bash
```

```bash
```

Function

Return the smaller of x and y

```bash
```

```bash
```

Return the absolute value of x - y

```bash
```

```bash
```

Calculate x / y, but round up the result.

```bash
```

```bash
```

Return the value of a base raised to a power

```bash
```

```bash
```

Get a nearest lower integer Square Root for x. Given that this function can only operate with integers, it is impossible to get perfect (or precise) integer square root for some numbers.

Example:

In integer math, one of the possible ways to get results with more precision is to use higher values or temporarily multiply the value by some bigger number. Ideally if this is a square of 10 or 100.

Example:

```bash
```

```bash
```

Try to convert a u128 to a u8 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u16 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash
```

```bash
```

```bash
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (inclusive)

```bash
```

```bash
```

Function

Return the absolute value of x - y

```bash
```

```bash
```

Calculate x / y, but round up the result.

```bash
```

```bash
```

Return the value of a base raised to a power

```bash
```

```bash
```

Get a nearest lower integer Square Root for x. Given that this function can only operate with integers, it is impossible to get perfect (or precise) integer square root for some numbers.

Example:

In integer math, one of the possible ways to get results with more precision is to use higher values or temporarily multiply the value by some bigger number. Ideally if this is a square of 10 or 100.

Example:

```bash
```

```bash
```

Try to convert a u128 to a u8 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u16 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash
```

```bash
```

```bash
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f to each number from start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (inclusive)

```bash
```

```bash
```

Function

Calculate x / y, but round up the result.

```bash
```

```bash
```

Return the value of a base raised to a power

```bash
```

```bash
```

Get a nearest lower integer Square Root for x. Given that this function can only operate with integers, it is impossible to get perfect (or precise) integer square root for some numbers.

Example:

In integer math, one of the possible ways to get results with more precision is to use higher values or temporarily multiply the value by some bigger number. Ideally if this is a square of 10 or 100.

Example:

```bash
```

```bash
```

Try to convert a u128 to a u8 . Returns None if the value is too large.

```bash
```

```bash
```

```

Try to convert a u128 to a u16 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash

```

```bash

```

```bash

```

```bash

```

Maximum value for a u128

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (exclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (inclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash

```

```bash

```

Function

Return the value of a base raised to a power

```bash

```

```bash

```

Get a nearest lower integer Square Root for x. Given that this function can only operate with integers, it is impossible to get perfect (or precise) integer square root for some numbers.

Example:

In integer math, one of the possible ways to get results with more precision is to use higher values or temporarily multiply the value by some bigger number. Ideally if this is a square of 10 or 100.

Example:

```bash

```

```bash

```

Try to convert a u128 to a u8 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u16 . Returns None if the value is too large.

```bash

```

```bash
```

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash
```

```bash
```

```bash
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (exclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash

```

```bash

```

Function

Get a nearest lower integer Square Root for x. Given that this function can only operate with integers, it is impossible to get perfect (or precise) integer square root for some numbers.

Example:

In integer math, one of the possible ways to get results with more precision is to use higher values or temporarily multiply the value by some bigger number. Ideally if this is a square of 10 or 100.

Example:

```bash

```

```bash

```

Try to convert a u128 to a u8 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u16 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash

```

```bash

```
```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash
```

```bash
```

```bash
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash
```

```bash
```

Function

Try to convert a u128 to a u8 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u16 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash
```

```bash
```

```bash
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (exclusive)

```bash
```

```bash
```

```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (inclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash

```

```bash

```

Function

Try to convert a u128 to a u16 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash

```

```bash

```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash

```

```bash

```

```bash

```
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (inclusive)

```bash
```

```bash
```

Function

Try to convert a u128 to a u32 . Returns None if the value is too large.

```bash
```

```bash
```

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash
```

```bash
```

```bash
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from f to each number from start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (exclusive)

```bash
```

```bash
```

Loops applying f to each number from 0 to f to each number from 0 to stop (inclusive)

```bash
```

```bash
```

Function

Try to convert a u128 to a u64 . Returns None if the value is too large.

```bash
```

```bash
```

```bash
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash
```

```bash
```

Function

```bash
```

```bash
```

Maximum value for a u128

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash
```

```bash
```

Macro function

Maximum value for a u128

```bash
```

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (exclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (inclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash

```

```bash

```

Macro function

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (exclusive)

```bash

```

```bash

```

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (inclusive)

```bash

```

```bash

```
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash
```

```bash
```

Macro function

Loops applying f t o e a c h n u m b e r f r o m f to each number from f t oe a c hn u mb er f ro m start to $stop (inclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash
```

```bash
```

Macro function

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (exclusive)

```bash
```

```bash
```

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash
```

```bash
```

Macro function

Loops applying f t o e a c h n u m b e r f r o m 0 t o f to each number from 0 to f t oe a c hn u mb er f ro m 0 t o stop (inclusive)

```bash
```

```bash
```