```
     ...: ran = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 11294000.75
The final decimal number is: 11294000.5
Precision BSE: 0.25
Precision IEEE754: 0.25
0.0

In [171]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 94749.8974609375
The final decimal number is: 94749.896484375
Precision BSE: 0.0009765625
Precision IEEE754: 0.0009765625
0.0

In [172]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 1438.26318359375
The final decimal number is: 1438.2626953125
Precision BSE: 0.00048828125
Precision IEEE754: 0.00048828125
0.0

In [173]: import
```

```
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 0.35101931542158127
The final decimal number is: 0.35101930797100067
Precision BSE: 7.450580596923828e-09
Precision IEEE754: 7.450580596923828e-09
0.0

In [174]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
     ...: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 68.28993892669678
The final decimal number is: 68.28993797302246
Precision BSE: 9.5367431640625e-07
Precision IEEE754: 9.5367431640625e-07
0.0
The final decimal number is: 5.428400695323944
The final decimal number is: 5.428400635719299
Precision BSE: 5.960464477539063e-08
Precision IEEE754: 5.960464477539063e-08
0.0

In [175]: import
```

```
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:             =      -
     ...:                 =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 125.6387710571289
The final decimal number is: 125.63876342773438
Precision BSE: 7.62939453125e-06
Precision IEEE754: 7.62939453125e-06
0.0

In [176]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:             =      -
     ...:                 =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 0.44505204260349274
The final decimal number is: 0.44505202770233154
Precision BSE: 1.4901161193847656e-08
Precision IEEE754: 1.4901161193847656e-08
0.0

In [177]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:             =      -
     ...:                 =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 469.4259605407715
The final decimal number is: 469.4259567260742
Precision BSE: 3.814697265625e-06
Precision IEEE754: 3.814697265625e-06
0.0
```

```
In [178]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 7779481
The final decimal number is: 7779480
Precision BSE: 1
Precision IEEE754: 1.0
0.0
```

```
In [179]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 462776.35546875
The final decimal number is: 462776.3515625
Precision BSE: 0.00390625
Precision IEEE754: 0.00390625
0.0
```

```
In [180]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
     ...: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
     ...:
The final decimal number is: 578.0003051757812
The final decimal number is: 578.000244140625
Precision BSE: 6.103515625e-05
Precision IEEE754: 6.103515625e-05
0.0
The final decimal number is: 0.14682624489068985
The final decimal number is: 0.14682623744010925
Precision BSE: 7.450580596923828e-09
Precision IEEE754: 7.450580596923828e-09
0.0

In [181]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 207360.509765625
The final decimal number is: 207360.5078125
Precision BSE: 0.001953125
Precision IEEE754: 0.001953125
0.0

In [182]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 207.4191131591797
The final decimal number is: 207.41909790039062
Precision BSE: 1.52587890625e-05
Precision IEEE754: 1.52587890625e-05
```

```
0.0

In [183]: list range 31
Out[183]:
[0,
 1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
 19,
 20,
 21,
 22,
 23,
 24,
 25,
 26,
 27,
 28,
 29,
 30]

In [184]: len list range 31
Out[184]: 31

In [184]:

In [184]:

In [185]: import
     ...: import       as
     ...:     = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 805699.5546875
The final decimal number is: 805699.546875
```

```
Precision BSE: 0.0078125
Precision IEEE754: 0.0078125
0.0

In [186]: import
     ...: import       as
     ...:    = ''
     ...: for   in range 31
     ...:       =       .        0  1  # decide on a k each time the loop runs
     ...:         += str
     ...:      =                       + '1'
     ...:      =                       + '0'
     ...:              =      -
     ...:                  =   .                0  -   .                0
     ...: print "Precision BSE: " + str
     ...: print "Precision IEEE754: " + str
     ...:      =              -
     ...: print
The final decimal number is: 299.33471298217773
The final decimal number is: 299.33470916748047
Precision BSE: 3.814697265625e-06
Precision IEEE754: 3.814697265625e-06
0.0

In [187]:
Out[187]: 299.33470916748047

In [188]:
Out[188]: 299.33471298217773

In [189]:   .          101010 0
Out[189]: 101009.99999999999

In [190]: ?numpy.nextafter
Object `numpy.nextafter` not found.

In [191]: ?numpy.nextafter
Object `numpy.nextafter()` not found.

In [192]: ?np.nextafter
Object `np.nextafter()` not found.

In [193]: ?ny.nextafter
Object `ny.nextafter()` not found.

In [194]: ?np.nextafter
Object `np.nextafter()` not found.

In [195]: ?np.nextafter
Call signature:  np.nextafter(*args, **kwargs)
Type:            ufunc
String form:     <ufunc 'nextafter'>
File:            ~/anaconda3/envs/scipro/lib/python3.8/site-packages/numpy/
__init__.py
Docstring:
nextafter(x1, x2, /, out=None, *, where=True, casting='same_kind', order='K',
dtype=None, subok=True[, signature, extobj])

Return the next floating-point value after x1 towards x2, element-wise.
```

```
Parameters
----------
x1 : array_like
    Values to find the next representable value of.
x2 : array_like
    The direction where to look for the next representable value of `x1`.
    If ``x1.shape != x2.shape``, they must be broadcastable to a common
    shape (which becomes the shape of the output).
out : ndarray, None, or tuple of ndarray and None, optional
    A location into which the result is stored. If provided, it must have
    a shape that the inputs broadcast to. If not provided or None,
    a freshly-allocated array is returned. A tuple (possible only as a
    keyword argument) must have length equal to the number of outputs.
where : array_like, optional
    This condition is broadcast over the input. At locations where the
    condition is True, the `out` array will be set to the ufunc result.
    Elsewhere, the `out` array will retain its original value.
    Note that if an uninitialized `out` array is created via the default
    ``out=None``, locations within it where the condition is False will
    remain uninitialized.
**kwargs
    For other keyword-only arguments, see the
    :ref:`ufunc docs <ufuncs.kwargs>`.

Returns
-------
out : ndarray or scalar
    The next representable values of `x1` in the direction of `x2`.
    This is a scalar if both `x1` and `x2` are scalars.

Examples
--------
>>> eps = np.finfo(np.float64).eps
>>> np.nextafter(1, 2) == eps + 1
True
>>> np.nextafter([1, 2], [2, 1]) == [eps + 1, 2 - eps]
array([ True,  True])
Class docstring:
Functions that operate element by element on whole arrays.

To see the documentation for a specific ufunc, use `info`.  For
example, ``np.info(np.sin)``.  Because ufuncs are written in C
(for speed) and linked into Python with NumPy's ufunc facility,
Python's help() function finds this page whenever help() is called
on a ufunc.

A detailed explanation of ufuncs can be found in the docs for :ref:`ufuncs`.

Calling ufuncs:
===============

op(*x[, out], where=True, **kwargs)
Apply `op` to the arguments `*x` elementwise, broadcasting the arguments.

The broadcasting rules are:

* Dimensions of length 1 may be prepended to either array.
* Arrays may be repeated along dimensions of length 1.
```

```
Parameters
----------
*x : array_like
    Input arrays.
out : ndarray, None, or tuple of ndarray and None, optional
    Alternate array object(s) in which to put the result; if provided, it
    must have a shape that the inputs broadcast to. A tuple of arrays
    (possible only as a keyword argument) must have length equal to the
    number of outputs; use None for uninitialized outputs to be
    allocated by the ufunc.
where : array_like, optional
    This condition is broadcast over the input. At locations where the
    condition is True, the `out` array will be set to the ufunc result.
    Elsewhere, the `out` array will retain its original value.
    Note that if an uninitialized `out` array is created via the default
    ``out=None``, locations within it where the condition is False will
    remain uninitialized.
**kwargs
    For other keyword-only arguments, see the :ref:`ufunc docs <ufuncs.kwargs>`.

Returns
-------
r : ndarray or tuple of ndarray
    `r` will have the shape that the arrays in `x` broadcast to; if `out` is
    provided, it will be returned. If not, `r` will be allocated and
    may contain uninitialized values. If the function has more than one
    output, then the result will be a tuple of arrays.
```

```
In [196]:
```