Math 271.1: Exercise 1 (#1)

INSTRUCTION:

Starting from x = 1, repeatedly set x = x/2 until 1 + x == 1.
Record the last x for which 1 + x > 1.
Compare this value with eps.

Main Data Structure / Variables

- ***Floating-point***:
  - **x**: initialized with 1
  - **last_x**: keeps track of the last value of x before it fails the condition and jumps out the loop, should be initialized to zero or None
  - **eps64**: holds the machine eps in double precision (default in python for float)

Pseudocode

- Generate the variables: x, last_x, eps64
- Use '**while-loop**' for the iteration, since we should at least have 1 iteration and we know the logic when it should stop
  - In every iteration, track last_x and divide the x into half
  - Once it fails the condition, jump outside loop and compare with the eps64 value

Step by step breakdown of logic:

Initialize variables:

```
x = 1.0
last_x = None
```

```
# In python, we can get built-in value for eps
import numpy as np

eps64 = np.finfo(float).eps
print("float64 eps:", eps64) ## since float64 is the default used by pyth
```

```
float64 eps: 2.220446049250313e-16
```

Iteratively half the x and track the last x

Evaluate whether (1.0 + x) is in fact equal to 1.0 or not. If not, continue adding the previous x value halved, Otherwise, stop and jump outside the loop.

```
while 1.0 + x != 1.0:
    last_x = x
    x = x / 2.0
```

By the time the loop ends, x is half of the latest value of x (now last_x):

```
print(last_x) #previous value
print(x) # current value
```

```
2.220446049250313e-16
1.1102230246251565e-16
```

Comparison of Machine Eps vs Last X value:

When checking equality, '==' should not be used because it is particularly dangerous for floating-point arithmetic. It will result to rounding error.
To be sure we can use Numpy's `allclose`, which checks if two values matches within certain tolerance

```
if last_x == eps64:
    print('Equal')
else:
    print('Different by: ', last_x - eps64)
```

```
Equal
```

Entire Code:

```
import numpy as np

x = 1.0
last_x = None
eps = np.finfo(float).eps

while 1.0 + x != 1.0:
    last_x = x
    x = x / 2.0
```

```
print('Value of x: ', x)
print("Value of eps:", eps)
print('Last value of x:', last_x)

abs_error = abs(last_x - eps)

print(f"\nAbsolute error between last x and eps: {abs_error:.5f}")
print(f"Approximately the same? {np.allclose(last_x, eps)}")
```

```
Value of x:  1.1102230246251565e-16
Value of eps: 2.220446049250313e-16
Last value of x: 2.220446049250313e-16

Absolute error between last x and eps: 0.00000
Approximately the same? True
```