CSCI1200 Python Practice Exercises (Binary Search)

These are self-practice exercises for beginners. These exercises are easier than the lab and exam questions. The purpose of these exercises is to ensure that you have mastered the minimum basics of programming.
You are asked to work out these exercises by hand, rather than implementing them.

Binary Search

1. Number of iterations
2. start and end
3. the conditions
4. // vs /

Practice Question 2 Binary search integer cube root
Write a Python function `integer_cube_root_binary_search` that takes in a positive integer x and return the integer cube root of x if it exists, return None otherwise. Your function should implement binary search to find the integer cube root.
Requirement: You are required to find the integer cube root using binary search.
```
Step-By-Step Instructions:
1. start, end=0,x
2. Run an iteration as long as start<=end. At each iteration,
   a. let m be the integer that lies in the middle of start and end.
   b. If  m**3 is equal to x, then return m.
   c. Else if m**3< x, then set start to m+1.
   d. Else (m**3> x), then set end to m-1.
3. Return None
```

```
def integer_cube_root(x):

  start, end = 0, x
  iter = 0
  while start<=end:
    iter+=1
    m = (start+end)//2
    if m**3==x:
      return m
    elif m**3<x:
      start = m+1
    else:
      end = m-1
  print("Number of iteration", iter)
  return None

## Practice Question 3: Binary search cube root
Write a Python function `cube_root_binary_search` that takes in a positive number $x$ and return the cube root of $x$. Your function should implement binary search to find the cube root.
Remark: It is sufficient to return the value $r$ such that $r^3$ is "close" to $x$ within 0.01, that is $|r^3-x|<0.01$.
Requirement: You are required to find the cube root using binary search.
```

Step-By-Step Instructions:
1. start, end=0,x
2. Run an iteration as long as start<=end. At each iteration,
   a. Let m be the middle of start and end.
   b. If abs(m**3-x)<0.01, then return m.
   c. Else if m**3<x, then set start to m.
   d. Else (m**3>x), then set end to m.

```
## Binary search on cube_root
def cube_root(x):


  epsilon = 0.01
  start, end = 0, x
```

```
while(start<=end):
    m  = (start+end)/2
    if abs(m**3-x)<epsilon:
      return m
    elif m**3<x:
      start = m
    else:
      end = m
```