CSCI 1200 Midterm2 Practice

By: Dr. Ming Ming Tan.

Name: _____ Scores: /30

Instructions

This is a closed-book exam, without notes, books, or electronic devices.
A document (Python cheatsheet) that provides basic guide to Python syntax is attached at the back of this document.
The duration of the exam is 50 minutes.
Take your time to read the statements carefully before trying to answer them.
When writing code, make sure your special punctuation characters are legible, and your lowercase and uppercase letters are easy to distinguish.
There are in total 10 questions and one bonus question. The weight of the Midterm 2 is 15%.
The bonus question contributes an additional 3 points that counts towards the final score.
Please only attempt the bonus question (optional) after completing all the other questions.
Please note that the number of lines given to you in the code block does not reflect the number of lines of code you need to write.

Question 1a

Consider the following function, where `L` is assumed to be a list of integers.

```
def test(L):
  for i in range(len(L)):
      if L[i]%2==1:
         return i
  return -1
```

What is the value of
(i) `test([1, 2, 3, 4, 5])`
(ii) `test([1, 3, 5])`
(iii) `test([2,3,4])`

Question 1b

Consider the `test` function in Question 1a.
Which of the following statements are correct. Select all that apply.
(i) If the input list `L` is empty, executing `test(L)` will give an error.
(ii) If the input list `L` contains only even integers, then `test(L)` will always return `-1`.
(iii) The function `test` takes an input list of integers and returns the index of the first occurrence of an odd integer.

Question 2a

Consider the following function.

```
def test(n):
  result = ""
  while n>0:
    result += str(n%2)
    n = n//2
```

```
  return result[::-1]
```

What is the output of:
(i) `test(4)`
(ii) `test(5)`
(iii) `test(6)`

Question 2b

Consider the `test` function in Question 2a.
Which of the following statements are correct. Select all that apply.
(i) If `n` is 0, the output is an empty string.
(ii) For positive integer `n`, `test(n)` returns the binary representation of `n` as string.
(iii) For positive integer `n`, `test(n)` returns the binary representation of `n` as string in reversed order.

Question 3a

What is the finite binary representation of
(i) 0.1
(ii) 0.5
(iii) 0.125

Question 3b

Consider the following code:

```
def verify_cube_root(r,n):
  return r**3==n
```

The output of `verify_cube_root(0.1, 0.001)` is `False`. Why?

Question 4a

Let `n` be a positive integer. We say that `n` is a prime if `n` is divisible only by 1 and `n`. Consider the following code for checking if a positive integer `n` is a prime.

```
def isPrime(n):
  if n%2==0:
    return False
  for d in range(3, n, 2):
     if n%d==0:
        return False
  return True
```

(i) What is the value of `isPrime(2)`?
(ii) What is the value of `isPrime(11)`?
(iii)) Note that 2 is a prime integer because the only divisor of 2 is 1 and itself. Hence, the function `isPrime` defined above is wrong. Complete the following function so that the `isPrime` function returns `True` when the input `n` is 2.

```
def isPrime(n):
  if _____________:
    return True
  if n%2==0:
    return False
  for d in range(3, n, 2):
     if n%d==0:
        return False
  return True
```

Question 4b

Write a function `first_divisor(n)` that takes a positive integer `n` and returns the smallest divisor of `n` that is not `1` and `n`. If such a divisor doesn't exist (when `n` is prime), then returns -1.
Hint:

1. If `n` is 2, what should the function return ?
2. If `n` is greater than 2, run an iteration on `p` from 2 to `n`-1 to check if `p` divides `n`. If yes, return `p`.
3. After the iteration is completed (i.e., no value of `p` from 2 to `n`-1 divides `n`), what should the function return ?

```
def first_divisor(n):
```

Question 5a

Complete the function `integer_square_root(n)` which takes an input positive integer n and returns the largest integer r such that $r^2 <= n$, implemented using the following algorithm:
Algorithm:
Run an iteration on `r` from 0 to `n`. At each iteration, do the following

a. If square of `r` is equal to `n`, then return ____.
b. Else if square of `r` is greater than `n`, then return _____.

For examples:

1. `integer_square_root(4)` should return 2
2. `integer_square_root(15)` should return 3

```
def integer_square_root(n):
  for r in ____________:
```

```
    if _______________:
        return ________
    elif _____________:
        return __________
```

Question 5b

Implement the `integer_square_root(n)` function define in Question 5a using binary search.
Hint:

1. Set `start` to 0 and `end` to `n`.
2. Set `result` = 0.
3. Run an iteration as long as `start` is less than or equal to `end`.
   In each iteration, do the following:
   a. Take the middle `m` of `start` and `end`.
   b. If square of `r` is equal to `n`, then return `m`.
   c. Else if square of `r` is less than `n`, set `result` to be equal to `m`, and set `start` to ___.
   d. Else, set `end` to _.
4. Return `result`.

```
def integer_square_root(n):
```
