Carnegie Mellon University in Qatar

Principles of Computing

15-110 - Fall 2018

Homework Assignment 3

Out: September 30, 2018

Due: 11:00 AM, October 06, 2018

2pts 1. What will be the result of attempting to run the following program? Please try to run it in your head before testing it on your computer.

```
counter = 0
for i in range(10):
  j = 0
  while j < 10:
    j = j + 1
  if j > i:
    break
  if j == i:
    counter = counter + 1
    continue
print(counter)
```

Select the one correct answer.

- ◯ The program will fail to run.
- ◯ The program will not terminate normally.
- ◯ The program will write 10 to the standard output.
- ◯ The program will write 0 to the standard output.
- ◯ The program will write 9 to the standard output.

15pts 2. Write a program that includes the following functions:

1. A function that sums the first n counting numbers: $1 + 2 + 3 + \ldots + n$, using only a `for` statement.
2. A function that sums the first n counting numbers: $1 + 2 + 3 + \ldots + n$, using only a `while` statement.
3. A function that sums the first n odd numbers: $1 + 3 + 5 + \ldots + 2n - 1$, using only a `for` statement.
4. A function that sums the first n odd numbers: $1 + 3 + 5 + \ldots + 2n - 1$, using only a `while` statement.
5. A function that sums a series of numbers entered by the user until the value 999 is entered. Note: 999 should not be part of the sum.
6. A function that accepts only strings with sizes less than 20 characters and prints the characters of any accepted string on one line, all separated by spaces (e.g., if the string "programming" is passed to the function, the function will accept it and print:
   p r o g r a m m i n g
   on one line).
7. A function that returns the number of times a whole number n can be divided by 2 (using integer division) before reaching 1.

Call all your functions in your program and verify that they all work correctly. Submit your program in one Python module named `SetOfFunctions.py`.

16pts 3. A positive whole number $n > 2$ is prime if no number between 2 and $\sqrt{n}$ (inclusive) evenly divides n.

1. Write a program that accepts a value of n as input and determines if the value is prime. If n is not prime, your program should quit as soon as it finds a value that evenly divides n. Submit this program in a separate Python module named `Prime1.py`.
2. Modify your `Prime1.py` program to find every prime number less than or equal to n. Submit this program in a separate Python module named `Prime2.py`.
3. The *Goldbach* conjecture asserts that every even number is the sum of two prime numbers. Write a program that gets a number from the user, checks to make sure that it is even, and then finds two prime numbers that add up to it. Submit this program in a separate Python module named `Prime3.py`.

7pts 4. The greatest common divisor (GCD) of two values can be computed using *Euclid*'s algorithm. Starting with the values m and n, we repeatedly apply the formula: $n, m = m, n\%m$ until m is 0. At that point, n is the GCD of the original m and n. Write a program that finds the GCD of two numbers using this algorithm. Submit this program in a separate Python module named `GCD.py`.

10pts 5. The *Syracuse* (also called *Collatz or Hailstone*) sequence is generated by starting with a natural number and repeatedly applying the following function until reaching 1:

$$syr(x) = \begin{cases} x/2 & \text{if } x \text{ is even} \\ 3x+1 & \text{if } x \text{ is odd} \end{cases}$$

For example, the *Syracuse* sequence starting with 5 is: $5, 16, 8, 4, 2, 1$. It is an open question in mathematics whether this sequence will always go to 1 for every possible starting value.

Write a program that gets a starting value from the user and then prints the Syracuse sequence for that starting value.