CSC1001: Introduction to Computer Science

Programming Methodology

Assignment 1

Assignment description:
This assignment will be worth 7% of the final grade.

You should write your code for each question in a .py file (please name it using the question name, e.g. q1.py). Please pack all your .py files into a single .zip file, name it using your student ID (e.g. if your student ID is 123456, then the file should be named as 123456.zip), and then submit the .zip file via Moodle.

Please also write a text file, which provide the details about how to run your code for each question. The text file should be included in the .zip file as well.

Please note that, the teaching assistant may ask you to explain the meaning of your program, to ensure that the codes are indeed written by yourself. Please also note that we may check whether your program is too similar to your fellow students' code using Moodle.

This assignment is due on 5:00PM, 11 March (Sunday). For each day of late submission, you will lose 10% of your mark in this assignment. If you submit more than three days later than the deadline, you will receive zero in this assignment.

Question 1 (*10% of this assignment*): Suppose you want to deposit a certain amount of money into a savings account with a fixed annual interest rate. What amount do you need to deposit in order to have $5,000 in the account after three years? The initial deposit amount can be obtained using the following formula:

$$initialDepositAmount = \frac{finalAccountValue}{(1 + annualInterestRate)^{numberOfYears}}$$

Write a program that prompts the user to enter final account value, annual interest rate in percent, and the number of years, and displays the initial deposit amount. A sample run of your program:

```
Enter the final account value:1000
Enter the annual interest rate:4.25
Enter the number of years:5
The initial value is: 812.1190197993631
>>>
```

Question 2 (*15% of this assignment*): Write a program that prompts the user to enter an integer and displays the number in reverse order. Here is a sample run:

```
Enter an integer: 3125 Enter
3
1
2
5
```

Question 3 (*15% of this assignment*): Write a program to allow a user to input a number m, and then use a while loop to find the smallest integer n such that n^2 is greater than m. For example, if the user inputs m = 10, your program should output n = 4.

Question 4 (*15% of this assignment*): Write a program to allow a user to input a number N, and print a table with N rows and 3 columns. In the *m*th row, your program should output three numbers: m, m+1, and m^{m+1}. For example, when the user inputs N = 5, your program should output the following:

```
m      m+1    m**(m+1)
1      2      1
2      3      8
3      4      81
4      5      1024
5      6      15625
```

Your program should be robust enough to handle the possible improper inputs (e.g. the user inputs a negative N).

Question 5 (*20% of this assignment*): Write a program to allow a user to input an integer N, and print all the prime numbers which are smaller than N. For example, when the user inputs N = 10, your program should output

```
The prime numbers smaller than 10 include:
2 3 5 7
```

Your program should output at most 8 prime numbers in each line. Your program should also be robust enough to handle the possible improper inputs (e.g. the user inputs a string).

Question 6 (*25% of this assignment*): Given a function f(x), and a real interval [a, b], the numerical integration of f(x) over interval [a, b] can be calculated as:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} \frac{b-a}{n} f\left(a + \frac{(b-a)}{n} \times (i - 1/2)\right) \quad (1)$$

In equation (1), n represents the number of sub-intervals into which the interval [a, b] will be divided; and it controls the accuracy of numerical integration.

Write a program to allow the user to specify a trigonometric function f (f can only be sin, cos or tan), and input the interval end points a, b and number of sub-intervals n. Your program should then calculate the numerical integration of f over [a, b] using equation (1), and output the result. Your program should be robust enough to handle possible improper inputs (e.g. the user inputs a floating point number as n).

Python has built-in trigonometric functions. To call them, use the following statement in your program to import them from the math package:

```
>>> from math import sin
>>> from math import cos
>>> from math import tan
```

You can then invoke the trigonometric functions like the following examples:

```
>>> sin(1)
0.8414709848078965
>>> cos(3.1415)
-0.9999999957076562
>>> tan(0)
0.0
```

For more details about the math package, please visit the following link:
https://docs.python.org/3/library/math.html#math.sin