IT1007 Introduction to Programming with Python and C Lab Exercise 06

Submission instructions:

1. There are two parts in this lab. For Part A, you have to submit within the same day of your lab session. For your Part B, you have six days to work on it. (E.g. if your TLab is on Monday, then your deadline is the coming Sunday midnight.)
2. Complete your code using the skeleton files provided, then **test your code on your computer first** before submitting to Coursemology.
3. To submit your code on Coursemology, click on "Labs" in the sidebar followed by the appropriate "Attempt" button.
4. **Copy ONLY the required function** from your completed skeleton file into the Coursemology code window.
5. Click "Run Code" to test that your function works on Coursemology.
6. Click "Finalise Submission" to submit your code for the **ENTIRE Part A/B**. **You will not be able to amend your code after you have finalised your submission.**
7. You must name your functions exactly as the questions state.

Failure to follow each of the instruction will result in 10% deduction of your marks.

Part A (Deadline: Same day of your TLab)

Questions 1 and 2: Multiple Choice Statements (5 Marks)

Question 1: Choose the variable names that are not allowed (List ALL answers!):

a) _!T!oo7
b) ILIKEIT1007
c) i-love-c
d) 1007_IT_I_LIKE
e) _IT_1007_is_my_favourite

Question 2: Which of the following options will be the output? (_ represents a blank space)

```
float var = 12345.6789123;
printf("%12.3f", var);
```

a) _ _ _ _ 12345.679
b) 12345.678912
c) _ _ _ 12345.678
d) _ _ _ _ _ _ _ 12345.678
e) _ _ _ 12345.679

Question 3: Debugging (15 Marks)

Correct the code shown below and submit the edited code on Coursemology.

```
#include <stdio.h>

int main () {

    int n;
    int counter;
    scanf("%s", n);

    if (n <= 1) {
        printf('Please input a value greater than 1\n');
        return 0;
    }

    for (counter, counter < n, counter +) {
        if n % counter == 0:
            printf("Not prime\n");
            return 0;
    }

    printf("Prime\n");
    return 0

}
```

Question 4: Code Translation [20 Marks]

IMPORTANT: You are required to only use scanf or scanf_s for reading inputs. This is to make you more familiar with type declaration/conversion when reading inputs. **Do not use** other functions.

Translate the provided Python code into one that can be run by the C compiler and provides the same results. Some sample runs are provided for your reference. **(Remember the \n in your final print statements!)**

```
def fibonacci():
    n = int(input("Please enter a number: "))
    term1 = 1
    term2 = 1

    if n == 0:
        print("Fibonacci number " + str(n) + "is 1")
    elif n == 1:
        print("Fibonacci number " + str(n) + "is 1")
    else:
        for counter in range (2,n+1):
            result = term1 + term2
            term1 = term2
            term2 = result
        print("Fibonacci number " + str(n) + " is " + str(result))
```

```
Please enter a number: 0
Fibonacci number 0 is 1

Please enter a number: 1
Fibonacci number 1 is 1

Please enter a number: 2
Fibonacci number 2 is 2

Please enter a number: 5
Fibonacci number 5 is 8

Please enter a number: 10
Fibonacci number 10 is 89

Please enter a number: 20
Fibonacci number 20 is 10946
```

Try entering an input number of 48. Then, run the program again and enter an input number of 49.
Is there anything strange that occurs? Why does this problem happen and how can you fix it?

Part B (Deadline: Day of TLab + 6 days)

Question 5: CPU Power Calculation (25 Marks)

The power consumed by a CPU, is approximately proportional to CPU frequency, and to the square of the CPU's operating voltage as given by the formula:

$$P = CV^2f$$

where C is the capacitance, f is the frequency and V is the voltage. This is referred to as a Dynamic Voltage Scaling (DVS) processor.

Let C take the value 0.00058 Farads. For every 10% increase in voltage, the frequency increases about 10 MHz in the design. The initial set of values are given in the table below. **Complete the table for the next 15 values.**

INITIAL VALUES

V = 3.5
f = 150

You should be able to obtain the initial value of P by yourself.

You program should print the table as shown below. Declare the variables correctly. You can use the loop of your choice. **Your results must be accurate up to 5 decimal points (yy.xxxxx) for V, P and % increase in P.**

(There are 10 "=" signs printed in each top and bottom row, and there are 5 spaces between each column)

==========
V(volts) = 3.50000 f(MHz) = 150 P(watts) = 1.06575
V(volts) = ? f(MHz) = ? P(watts) = ? % increase in P = ?
V(volts) = ? f(MHz) = ? P(watts) = ? % increase in P = ?
…
==========

A sample of the first few lines is shown below:

The output must end with the `\n` character (e.g. `"0\n"`).

EXAMPLE

Output:

```
==========
V(volts) = 3.50000     f(MHz) = 150     P(watts) = 1.06575
V(volts) = 3.85000     f(MHz) = 160     P(watts) = 1.37553     % increase in P = 29.06665
V(volts) = 4.23500     f(MHz) = 170     P(watts) = 1.76841     % increase in P = 28.56250
```

Important notes

1. In total, there should be 16 lines of values, with the first row being the base case.
2. You may notice some alignment issues when your V or P is >= 10. That is normal. **DO NOT PAD THE NUMBERS.**
3. Take note of the newline character after the bars at the top and bottom.

Question 6: Reversing Strings (15 Marks)

IMPORTANT: You are required to only use scanf or scanf_s for reading inputs. This is to make you more familiar with type declaration/conversion when reading inputs. **Do not use** other functions.

Read an input string (of not more than 30 characters) from the user and do the following:

a) Measure the length of the string, then print the string and its length

The output must end with the `\n` character (e.g. `"0\n"`).

EXAMPLE

Input:

```
abc
```

Output:

```
abc 3
```

b) Print the string in the **reverse** order.

EXAMPLE

Input:

```
abc
```

Output:

```
cba
```

HINT: Consider what library functions you may use to implement the logic for this question!

Question 7: Colour-Coding (20 Marks)

IMPORTANT: You are required to only use scanf or scanf_s for reading inputs. This is to make you more familiar with type declaration/conversion when reading inputs. **Do not use** other functions.

Design a program that takes in a char input from the user and demonstrate different behaviour based on the input character as stated in the rules below. This must be implemented with a **switch** statement, and the user's input and token generated should be printed as follows in the reference cases on the right:

User Input: 'R' or 'r'
Generate a random integer number in the range [0,5]

User Input: 'Y' or 'y'
Generate a random integer number in the range [6,10]

User Input: Any other input
Generate a random integer number in the range [11,15]

HINT: How can you make use of the modulo (%) operator to generate a random number within a certain range?

The output must end with the `\n` character (e.g. `"0\n"`).

Refer to coursemology for more examples.

EXAMPLE: User Input 'R'

Input:

```
R
```

Output:

```
3
```