

Computer Science 1001

Assignment 2, Part A

Fall 2022

Recommended Completion Date for Part A: Monday, October 10, 2022

Assignment 2 (Parts A and B) due by Monday, October 24, 2022 at 11:00 pm

Important notes:

- Each assignment will come in two parts (Part A and Part B – each covering various chapters/concepts), and several question solutions from each part must be submitted to the corresponding **Assignment** dropbox, available from the **Assessment** menu in Brightspace. Not all question solutions will be graded. Each week the course instructors will select a **single question from each part** to be graded by the markers. Feedback will be given only on those two select questions and your marks will be based on those two questions only. The other submissions will not be graded and you will not be told, ahead of time, which questions will be graded. In order to ensure full opportunity to achieve the best grade, you should submit all questions labelled as **(this question MAY BE GRADED)** below. The graded question will be identified in the feedback for the assignment. Some questions are labelled as **(this question will not be graded)** and hence you do not need to submit this question at all, but it is still a good idea to complete it to understand the concepts.
- While doing your assignment you can get help by posting your questions in the appropriate folder in the discussion forum to get help from your fellow students, the instructor and the instructional staff. You should also read other posts there since this may help with your questions.
- Completing the assignments will help you to gain hands on experience with coding in Python and understand the learnt concepts well enough so that you can apply the concepts to solve and code the solution for the given problems. All of this will help you in doing well on your tests and exam.
- Solutions to these assignments will not be posted but you will get feedback on a portion of your assignment submission.
- While coding solutions for the problems given below, keep in mind that on the quizzes/tests/exams you will also be marked on the following:
 - efficient solution of the problem. A given problem can be solved in number of different ways, but the best solution is the one that is efficient; i.e., the one that uses the right concepts in a very productive way.
 - including sufficient descriptive comments in your program. The person looking at your code should be able to understand how your code is solving the given problem even if the person reading your Python program does not know the Python language. In addition, the reader of your program should be able to understand what each variable represents.
 - labelling of input and output. All input/output should have a descriptive label so that the reader of your program understands what input is expected and what output the program has generated.
 - program style - consistent formatting and indentation of program statements, meaningful variable names (identifiers) and the use of constants (constant identifiers), where appropriate.
 - Note that sometimes, even though your code can generate the expected output, it may still not be correct as it may work for a specific data set and not for all valid data sets. Also, it may not use the programming concepts and best practices that we have emphasized in the course.Practicing these rules will build a good foundation for programming.
- This assignment is based on Chapter 4 of the text, excluding the graphics components from the chapter. Please use concepts from Chapters 1 – 4, including the following concepts:
 - counter-controlled repetition structures using
 - for loop with range function
 - while loop with conditional expressions that use the concept of a counter

- condition-controlled repetition structures using a while loop with conditional expressions that use the concepts of flags/sentinel values
- conditional expressions that could be
 - simple relational expressions
 - compound logical expressions
- nested repetition structures
- **IMPORTANT: DO NOT USE** the **break** (text page 138) or **continue** statements even though these are valid Python statements.
- Please use only the concepts/structures/functions, etc. that we cover in the course. Using advanced concepts/structures/functions that we do not cover in the course may cause you to lose marks since each assignment/problem is testing that you understand specific concepts.
- **Assignments are to be done individually. The solutions that you submit must be your own work.**
- **Posting of assignment questions to homework websites is not permitted.**

With this understanding please complete the following questions:

Question 1 (a) – (this question will not be graded)

Trace (by hand) the execution of the following program, by completing the trace table below, for any lines of code that are labelled with line numbers. The first line has already been filled in for you. Use the following input values (in this order, some inputs may not need to be used): 9 8 27 10 15

```

LIMIT = 3                                     #LINE 1
yes = 0                                       #LINE 2
no = 0                                       #LINE 3
for i in range(LIMIT, 0, -1):                #LINE 4
    isIt = int(input("Enter the input value: ")) #LINE 5
    if isIt % LIMIT == 0:                    #LINE 6
        yes += 1                             #LINE 7
    else:
        no += 1                               #LINE 8
print("Number of yes:", yes)                 #LINE 9
print("Number of no:", no)                   #LINE 10

```

Line No	LIMIT	yes	no	isIt	i	Loop Condition (i still in range? T/F)	If Condition (T/F)	Output
1	3							
2		0						
3			0					
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Question 1 (b) - (this question will not be graded)

Study and understand what the code given in **Question 1 (a)** is doing. Rewrite the code (from **Question 1 (a)**) using a `while` loop without introducing new variables. Trace (by hand) execution of the code you have developed by completing the trace table given below using the input values as given above: 9 8 27 10 15

For the trace, please completely renumber all lines of your rewritten code as necessary since new lines will have to be inserted.

Note: Your code should generate the same output as the code given in **Question 1 (a)**. **Do not introduce any new variables.**

Line No	LIMIT	yes	no	isIt	i	Loop Condition (i still in range? T/F)	If Condition (T/F)	Output
1	3							
2		0						
3			0					
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Question 2 - (this question MAY BE GRADED)

Call the files containing your solutions to the following, `Question2a.py` and `Question2b.py`, respectively.

a). Given the following `while` loop, rewrite it using a `for` loop.

```
i = 10
while i <= 190:
    print("The value of loop control variable i is", i)
    i += 10
```

b). Given the following `for` loop rewrite it using a `while` loop.

```
for i in range (55, -1, -3):
    print ("The value of loop control variable i is", i)
```

Question 3 - (this question MAY BE GRADED)

Write a Python program, in a file called `dragon_training.py`, to solve the following problem:

The Viking village of Berk, located on a remote island, is attacked frequently by dragons, which take livestock, damage property, and endanger lives of the villagers. Hiccup, son of the village chieftain, Stoick the Vast, befriends the rarest of the dragons, Night Fury. All his friends looked forward to having their own dragons to ride. With the help of Night Fury and Hiccup, most kids captured a dragon for themselves when the dragons thronged and attacked the village.

To qualify to fully have your dragon under your command, the young Vikings are trained by the great dragon trainer Hoark the Haggard. After the young Vikings have completed their schooling, the young Vikings are put to the test where each one needs to **pass 5 different trainings that test both their physical and mental limitations**. Each one is evaluated based on their 5 different tests by Hoark.

Given the number of participants, the name and the total score of all five tests (full points would be a total of 50 points for all 5 tests) for each participant, select and output the names and points of the participants who received points between 30 to 50 (those who passed the training).

For example, given, as input, the values 5, "Astrid", 35.80, "Hiccup", 50.0, "Wartihog", 25.25, "Speedifist", 30.75, "Fishlegs", 15.15

The input prompts, input and output should be as follows (input shown in **bold blue**):

```
Please enter the number of participants: 5
Please enter the name of the participant: Astrid
Please enter the total points: 35.80
Astrid passed the training with 35.80 points.
Please enter the name of the participant: Hiccup
Please enter the total points: 50.0
Hiccup passed the training with 50.00 points.
Please enter the name of the participant: Wartihog
Please enter the total points: 25.25
Please enter the name of the participant: Speedifist
Please enter the total points: 30.75
Speedifist passed the training with 30.75 points.
Please enter the name of the participant: Fishlegs
Please enter the total points: 15.15
```

Question 4 - (this question MAY BE GRADED)

Write a Python program, in a file called `sun_moon.py`, to solve the following problem:

Assume that the population of Sunland (e.g., 200 (million)) this year is higher than that of Moonland's population (e.g., 150 (million)). If the annual rate of growth in Sunland is 1% and that in Moonland is 20%, and if these rates remain constant, then how many years will it take for the population of Moonland to exceed that of Sunland. Ask the user for values of the two populations, in millions, and report the number of years that it will take for the population of Moonland to exceed the population of Sunland. All output should be appropriately labelled. Note that if the population of Moonland happens to already exceed that of Sunland, then the program should report that it will take 0 years.

Question 5 - (this question MAY BE GRADED)

Write a Python program, in a file called `too_count.py`, to solve the following problem:

Given several words, print each word and the number of times the word "too" occurs in the word (note that the comparison should be case-sensitive). You are NOT permitted to use any string functions, such as `find()`, `count()`, etc., for this question. You may use the built-in functions, `len()`, `input()` and `print()`, as well as the slice operator (`:`), looping structures and decision structures.

For example, given the following list of words (terminated by the word "Done" which is not part of the list):
toothacheandtoothfairy, Cartooning, footstooltoobarstool, Tomato, TATTOO

The input prompts, input and output should be as follows (input shown in **bold blue**):

```
Please enter a word or Done to terminate the program: toothacheandtoothfairy
too occurs 2 time(s) in the word toothacheandtoothfairy
Please enter another word or Done to terminate the program: Cartooning
too occurs 1 time(s) in the word Cartooning
Please enter another word or Done to terminate the program: footstooltoobarstool
too occurs 3 time(s) in the word footstooltoobarstool
Please enter another word or Done to terminate the program: Tomato
too occurs 0 time(s) in the word Tomato
Please enter another word or Done to terminate the program: TATTOO
too occurs 0 time(s) in the word TATTOO
Please enter another word or Done to terminate the program: Done
```