Fall 2020
Due Oct 2nd

Computer Science 1001
Problem Set #2

Section 001/002

Important Notes:

- Even though this, or any other subsequent, Problem Set is only graded based on whether or not a reasonable attempt was made for each problem, it is a good idea to complete each Problem Set by the due date and upload it to the appropriate dropbox in the course Brightspace shell to get credit for completing the Problem Set and see if you have understood the required concepts.
- While working on your Problem Set you can get help by posting your questions in the appropriate folder in the discussion forum. 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.
- Completing the Problem Sets 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.
- The files that you upload to the dropbox should be your source code (`.py`) files, as practiced in Lab 0, and any other requested solution files.
- While coding solutions for the problems given below, keep in mind that on the test/exam you will also be marked on the following:
  - Efficient solution of the problem. A given problem can be solved in a number of different ways, but the best solution is the one that is efficient; ie., 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 and 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.

  Practicing these rules will build a good foundation for programming.
- This Problem Set is based on Chapter 4 of the textbook, without the graphics components. Please use only concepts from Chapters 1–4 of the textbook.

- Rubrics/solution outlines for each Problem Set will be provided after the grades for the Problem Set have been released.

Full solutions will not be posted, however you may get help to complete your Problem Set if the rubrics/solution outlines are insufficient.

With this understanding please complete the following questions:

---

1. (a) Trace (by hand) 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): 7 1 2 5 6 7

```
LIMIT = 5                                                    #LINE 1
c = 0                                                        #LINE 2
d = 0                                                        #LINE 3
for x in range(1, LIMIT+1):                                  #LINE 4
    isThis = int(input("Enter the input value: ")) #LINE 5
    if isThis > LIMIT:                                       #LINE 6
        c += 1                                               #LINE 7
    else:
        d += 1                                               #LINE 8
print("Value of c is ", c)                                   #LINE 9
print("Value of d is ", d)                                   #LINE 10
```

Line #	LIMIT	c	d	isThis	x	x in range (T/F)	if condition (T/F)	Output
1	5							
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Note: You may submit this solution as a text document, or PDF.

(b) Study and understand what the code given in Question 1 is doing. Rewrite the code (from Question 1) using a `while` loop. Trace (by hand) execution of the code you have developed by completing the trace table given below using the input values as given above: `7 1 2 5 6 7`.

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. **Do not introduce any new variables.**

Line #	LIMIT	c	d	isThis	x	x in range (T/F)	if condition (T/F)	Output
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Note: You may submit your trace table as a text document, or PDF. You should also submit your revised code (.py file).

2. Write a Python program in a file called `consonants.py`, to solve the following problem using a nested loop. For each input word, replace each consonant in the word with a question mark (`?`). Your program should print the original word and a count of the number of consonants replaced. Assume that the number of words to be processed is not known, hence a sentinel value (maybe `"zzz"`) should be used. You can assume that each input word contains only letters (ie. no digits, symbols, etc.)

   Note: You are not permitted to use the `replace`, `find`, `index`, `rfind` or `rindex` methods for this question.

   Sample input/output:

```
Please enter a word or zzz to quit: Dramatics
The original word is: dramatics
The word without consonants is: ??a?a?i??
The number of consonants in the word are: 6

Please enter another word or zzz to quit: organic
The original word is: organic
The word without consonants is: o??a?i?
The number of consonants in the word are: 4

Please enter another word or zzz to quit: YELLOW
The original word is: yellow
The word without consonants is: ?e??o?
The number of consonants in the word are: 4

Please enter another word or zzz to quit: queueing
The original word is: queueing
The word without consonants is: ?ueuei??
The number of consonants in the word are: 3

Please enter another word or zzz to quit: zzz
```

3. Write, test and document a Python program, in a file called `PackagesOnTruck.py`, to solve the following problem.

   The post office is sending packages to the airport to be shipped to different zones in NL. Their regular transportation system is broken so they must load a single rental truck to transport the packages. The capacity of the rental truck is given, as input, in pounds. Given the weight of each package (in pounds, weight will vary per package), the package ID number, and the name of the zone (East, West, South, North) that the package needs to be sent to, count and display the number of packages that can be loaded on the truck without exceeding the weight capacity. Also, print the information

for each package that will be loaded onto the truck, the total number of packages bound for each zone, and the final count of all the packages on the truck. Note that if the weight of the next package would put the weight limit for the truck over the capacity, then that package should not be loaded on the truck and the program should stop accepting input.

Note: Your program should not store (e.g., in a list) all of the values entered, and should not use the `sum()` function. The information should be processed as it is being entered.

Sample input/output:

```
Please enter the weight limit of the truck: 1000

Please enter the weight of a package: 100
Please enter the package ID number: 15
Please enter the zone: East

The following package is on the truck:
Pkg No. = 15 , Zone = East , Pkg Wgt = 100.0

Please enter the weight of a package: 500
Please enter the package ID number: 11
Please enter the zone: West

The following package is on the truck:
Pkg No. = 11 , Zone = West , Pkg Wgt = 500.0

Please enter the weight of a package: 325
Please enter the package ID number: 12
Please enter the zone: East

The following package is on the truck:
Pkg No. = 12 , Zone = East , Pkg Wgt = 325.0

Please enter the weight of a package: 25
Please enter the package ID number: 18
Please enter the zone: South

The following package is on the truck:
Pkg No. = 18 , Zone = South , Pkg Wgt = 25.0

Please enter the weight of a package: 200

2 packages are on the truck for East zone.
1 packages are on the truck for West zone.
0 packages are on the truck for North zone.
```

```
1 packages are on the truck for South zone.

4 packages are loaded on the truck.
```