ITI1120 Winter 2018 - ASSIGNMENT 3

Read the instructions below carefully. The instructions must be followed. This assignment is worth **5%** of your grade. The assignment is due on **Mar 5th, 8:00am**. *No late assignments will be accepted.*

The goal of this assignment is to learn and practice (via programming) the concepts that we have learned so far. This includes lists and range function. You will also learn how to solve a bigger problem (making a game) by breaking it into smaller parts.

This assignment has two parts. In part 1 you will implement 3 small programs. In part 2, you will implement a card game. Put all the below four required documents into a folder called `a3_xxxxxx`, zip that folder and submit it as explained in lab 1. (In particular, the folder (and thus your submission) should have the following files:

`a3_Q1_xxxxxx.py,`
`a3_Q2_xxxxxx.py`
`a3_Q3_xxxxxx.py`
`a3_GAME_xxxxxx.py`

For each function that you design for this assignment you have to include docstrings that specify:
- type contract
- description about what the function does (while mentioning parameter names)
- preconditions, if any.

For this assignment, unlike previous two, you do not need to submit a text file as evidence that you tested your functions. But as always, you are strongly encourage to test all your functions in Python shell as you have learnt in the previous assignments.

As always, your programs must run without syntax errors.

```
****************************************************************************************************
```
PART 1 (20 points)
```
****************************************************************************************************
```
```
=====================
```
Question 1: (5 points)
```
=====================
```
Implement a Python function called `count_pos` that takes a list of numbers as input (parameter) and returns the number of elements of that list that are positive (> 0). Then, in the `main` portion of your program should ask the user to input the list, then it should call `count_pos` function with that list, and print the result. You may assume that the user will always enter at least two elements, but your function `count_pos` should work for all lists of numbers including those of length 0 and 1.

Examples of program runs:

Example 1:
```
Please input a list of numbers separated by commas: 1,2,0,3,0,4
There are 4 positive numbers in your list.
```

Example 2:
```
Please input a list of numbers separated by commas: 1,-2,9
There are 2 positive numbers in your list.
```

Here is a way to ask a user for a list:
```
s = input("Please input a list of numbers separated by commas: ")
L = list(eval(s))
```

If a user enters at least two numbers separated by commas, variable `L` will refer to that list.

```
=====================
```
Question 2: (5 points)
```
=====================
```
A run is a sequence of consecutive repeated values. Implement a Python function called `two_length_run` that takes a list of numbers as input and returns True if the given list has is at least one run (of length at least two), and False otherwise. Make sure the function is efficient (i.e. it stops as soon as the answer is known). Then, in the `main`, your program should ask the user to input the list, then it should call `two_length_run` function, and print the result. You may assume that the user will always enter at least two elements, but your function `two_length_run` should work for all lists of numbers including those of length 0 and 1.

Examples of program runs:

Example 1:
```
Please input a list of numbers separated by commas: 1,4,3,3,4
True
```

Example 2:
```
Please input a list of numbers separated by commas: 1,2,3,3,3,4,6,5
True
```

Example 3:
```
Please input a list of numbers separated by commas: 1,2,3,4,3,2
False
```

=====================
Question 3: (10 points)
=====================
As mentioned, a run is a sequence of consecutive repeated values. Implement a Python function called `longest_run` that takes a list of numbers and returns the length of the longest run. For example in the sequence: 2, 7, 4, 4, 2, 5, 2, 5, 10, 12, 5, 5, 5, 5, 6, 20, 1 the longest run has length 4. Then, in the `main` portion of your program should ask the user to input the list, then it should call `longest_run` function, and print the result. You may assume that the user will always enter at least two elements, but your function longest_run should work for all lists of numbers of length at least 1.

Examples of runs:

Example 1:
```
Please input a list of numbers separated by commas: 1,1,2,3,3,3,3,3,6,5
5
```

Example 2:
```
Please input a list of numbers separated by commas: 6,6,7,1,1,1,1,4, 1
4
```

Example 3:
```
Please input a list of numbers separated by commas: 6,2,4,8,6
1
```

****
PART 2 (80 points)
****

For this part, you will program a card game described here:

http://www.classicgamesandpuzzles.com/Old-Maid.html

You will implement the two player version. One player will be the computer (i.e. your program) and the other a user of your program. In what follows, let's refer to the computer player as Robot and user player as Human. You may assume that Robot will always deal the cards.

As part of this assignment I provided the file called `a3_GAME_xxxxxx.py`. Replace xxxxxx in the file name with your student number. You should open that file and run it to see what it does already. All of your code has to go inside of that file. The file already has some functions that are fully coded for you and other functions for which only docstrings and partial or no code are provided. Designing your program by decomposing it into smaller subproblems (to be implemented as functions) makes programming easer, less prone to errors and makes your code more readable.

No part of the given code can be changed. Your code must go into clearly indicated places. No code can be added to the main. You can design some extra functions of your own if you like.

Functions `make_deck`, `wait_for_player` and `shuffle_deck` are already fully coded for you.

You need to develop the remaining functions: `deal_cards`, `remove_pairs`, `print_deck`, `get_valid_input`, and `play_game.` The functions must meet the requirements as specified in their docstrings (and as implied by the example program runs below and in the video).

The main bulk of your code (the game playing part) will go into the function called `play_game`. That function should use/call the other functions that you are required to develop (i.e. `deal_cards`, `remove_pairs`, `print_deck`, `get_valid_input`).

When developing function `get_valid_input` you may assume that Human will enter integer when asked for an integer, but you may not assume that it will be in the correct range.

The function `get_valid_input` gets the input from Human about which face-down card of Robot it wants. When it is Robot's turn to play you must implement it such that Robot takes a random card from Human. Also recall that what Human calls 3rd card, for example, is in position/index 2 in Robot's deck (as it is represented by a list).

Study the example of the program run below carefully to understand how your program should behave. The behaviour of the program that you see in the run is required — all aspects of it.

Some suggestions:
1. Study the provided code and understand what it does and how it should be used.

2. Spend some time thinking about various parts of the game that need to be implemented. For example, it needs to be able to display Human's deck to Human, it needs to be able to ask Human for what card she wants, it needs to be able to remove pairs from either Human or Robot's deck … etc.

The provided functions do quite of bit of that job for you. In the next assignment, Assignment 4, you will need to be breaking the problem into small subproblems (to be implemented as functions) by yourself.

3. When you are coding individual functions recall that you can test each function in the shell without finishing the remaining functions. For example, when implementing function `remove_pairs` you can test it in the shell by typing something like:

```
>>> remove_pairs(['10?', '2?', '5◇', '6?', '9?', 'A◇', '10◇'])
```

The shell should display (with cards not necessarily in this order):
```
['2?', '5◇', '6?', '9?', 'A◇']
```

Thus you can code and test your functions one by one (without completing the other parts)

4. The game alternates between Robot and Human. Think about how you can represent whose turn it is to play, in your program. One way is to have a variable that you set to zero when it is Robot's turn and to one when it is Human's turn. You also need to figure out what to test to see if the game is over.

EXAMPLE RUNS OF YOUR GAME:


```
=============================================================================
RUN 1
=============================================================================


Hello. My name is Robot and I am the dealer.
Welcome to my card game!
Your current deck of cards is:

6? 3? 5◇ 7◇ 10♡ 2? 5? J♡ 5? Q? K? J? 10◇ 5♡ 4? 2♡ A◇ 10? A? 6♡ J? A♡ 3? 8? J◇ Q♡

Do not worry. I cannot see the order of your cards
Now discard all the pairs from your deck. I will do the same.

Press enter to continue.


*************************************************************
Your turn.
```

Your current deck of cards is:

7◇ 4? 8? 10◇ A? K?

I have 7 cards. If 1 stands for my first card and
7 for my last card, which of my cards would you like?
Give me an integer between 1 and 7: 5
You asked for my 5th card.
Here it is. It is 10?

With 10? added, your current deck of cards is:

7◇ 4? 8? 10◇ A? K? 10?

And after discarding pairs and shuffling, your deck is:

K? 7◇ A? 8? 4?

Press enter to continue.

****
My turn.

I took your 4th card.

Press enter to continue.

****
Your turn.

Your current deck of cards is:

K? 7◇ A? 4?

I have 5 cards. If 1 stands for my first card and
5 for my last card, which of my cards would you like?
Give me an integer between 1 and 5: 0
Invalid number. Please enter integer between 1 and 5: -2
Invalid number. Please enter integer between 1 and 5: -5
Invalid number. Please enter integer between 1 and 5: 3
You asked for my 3rd card.
Here it is. It is 7?

With 7? added, your current deck of cards is:

K? 7◇ A? 4? 7?

And after discarding pairs and shuffling, your deck is:

A? K? 4?

Press enter to continue.

****
My turn.

I took your 2nd card.

Press enter to continue.

```
************************************************************
Your turn.

Your current deck of cards is:

A? 4?

I have 3 cards. If 1 stands for my first card and
3 for my last card, which of my cards would you like?
Give me an integer between 1 and 3: 2
You asked for my 2nd card.
Here it is. It is Q◊

With Q◊ added, your current deck of cards is:

A? 4? Q◊

And after discarding pairs and shuffling, your deck is:

A? Q◊ 4?


Press enter to continue.

************************************************************
My turn.

I took your 2nd card.

Press enter to continue.

************************************************************
Your turn.

Your current deck of cards is:

A? 4?

I have 3 cards. If 1 stands for my first card and
3 for my last card, which of my cards would you like?
Give me an integer between 1 and 3: 1
You asked for my 1st card.
Here it is. It is A?

With A? added, your current deck of cards is:

A? 4? A?

And after discarding pairs and shuffling, your deck is:

4?


Press enter to continue.

************************************************************
My turn.

I took your 1st card.
```

Press enter to continue.

```
**********************************************************
```
Ups. You do not have any more cards
Congratulations! You, Human, win


```
============================================================================
```
RUN 2
```
============================================================================
```

Hello. My name is Robot and I am the dealer.
Welcome to my card game!
Your current deck of cards is:

J? 10? 7♡ K? 9♡ 3? 3◇ 7◇ J? 5? 5? 10? 10◇ 9? A? 6◇ Q♡ A◇ 10♡ 3♡ K? 4? 5◇ 4◇ 4♡ 6?

Do not worry. I cannot see the order of your cards
Now discard all the pairs from your deck. I will do the same.

Press enter to continue.

```
**********************************************************
```
Your turn.

Your current deck of cards is:

5? Q♡ 4? 3◇

I have 3 cards. If 1 stands for my first card and
3 for my last card, which of my cards would you like?
Give me an integer between 1 and 3: -1
Invalid number. Please enter integer between 1 and 3: 5
Invalid number. Please enter integer between 1 and 3: 1
You asked for my 1st card.
Here it is. It is 4?

With 4? added, your current deck of cards is:

5? Q♡ 4? 3◇ 4?

And after discarding pairs and shuffling, your deck is:

5? Q♡ 3◇

Press enter to continue.

```
**********************************************************
```
My turn.

I took your 2nd card.

Press enter to continue.

```
**********************************************************
```
Your turn.

Your current deck of cards is:

5? 3◇

```
I have 3 cards. If 1 stands for my first card and
3 for my last card, which of my cards would you like?
Give me an integer between 1 and 3: 3
You asked for my 3rd card.
Here it is. It is Q♡

With Q♡ added, your current deck of cards is:

5? 3◇ Q♡

And after discarding pairs and shuffling, your deck is:

5? Q♡ 3◇


Press enter to continue.

***********************************************************
My turn.

I took your 1st card.

Press enter to continue.

***********************************************************
Your turn.

Your current deck of cards is:

Q♡ 3◇

I have 1 cards. If 1 stands for my first card and
1 for my last card, which of my cards would you like?
Give me an integer between 1 and 1: 1
You asked for my 1st card.
Here it is. It is 3?

With 3? added, your current deck of cards is:

Q♡ 3◇ 3?

And after discarding pairs and shuffling, your deck is:

Q♡


Press enter to continue.

Ups. I do not have any more cards
You lost! I, Robot, win
```