Adam El M’Rabet *IMSAL Python Crash Course* *28/10/2020*

Game: LOTTO

This project will be your first step in the world of coding complex software. Up to now you have been working on solving small problems using basic algorithms which were necessary in order to build the mentality of the algorithmic way of thinking and solving problems. Some of the algorithms you have written are even present in almost every software. But up to now, these have been nothing more than small building bricks. It is time to combine all these bricks and to build our first fully operational program … ladies and gentleman … this is the LOTTO game !
Carefully read this file entirely before starting to code.

Start of the game

At the start the program asks the user to enter his name. Only if his name is correct (you choose yourself what the correct name is), you ask for a password. If both are correct, game starts.
So if the name or password are incorrect the program exits immediately.
Once the game starts, the program asks for an amount of money (in euros) that the user wants to put in the game. That amount should be saved.

Rules

The game is a kind of LOTTO guessing game using virtual money. At the start of the game, a list of 5 random numbers is generated. Each of those 5 random numbers is a number between 1 and 30. Carefully make sure that **all** the numbers of the list are different.

Then you ask the user to input a first guess (number). You then check if the number is inside the list of the 5 to guess numbers.

- If the number is indeed present in the list, the user’s invested money gets doubled and you print “Congratulations, you have now still X numbers to guess. Your current amount is XXX euro”. Then you ask the user again to guess a number and you make the same verification again.
  Caution! Every time a number is guessed that number should then be removed from the list. Otherwise we can guess the same number again and again and become millionaire.
- But if the guessed number wasn’t present in the list, then the user loses 10% of his money and the program prints: “Fail ! You have now only XXX euro left, and you still need to guess X numbers.”. Then you ask again for an input etc.

This fact of repeatedly asking the user to enter a new number again and again has to be repeated until either the user is bankrupt (no money anymore) or the user has found all the numbers.
Feel free and creative on the messages that you’d print to the user in case he wins/or loses the game.

Implementation

You should open LOTTO1.py (with pycharm). This file contains the to be completed by you code for the game. I have given more details of what you need to do there. I also provided you some code already. This project is meant for you to understand the general structure through building on already existing resources. So carefully read the code and the comments. In the next steps you'll slowly have to build everything on your own.

The code is split in two parts:

1) THE NEEDED FUNCTIONS

This part contains the functions needed for the algorithm. The first has been implemented by me, the rest is for you.

- **generate_new_list()**
  → This function returns a new list containing 5 numbers from 1 to 30.
  → I have already implemented it, but you **absolutely have** to understand it.
- **value_is_in_list(value, lst)**
  → This function checks whether a given number is present in a given list. This is a boolean function. This means that it should return either True (if the number is present in the list) or False (if it's not).
- **remove_element_from_lst(element, lst)**
  → This function returns a new list that contains all the elements of "lst", except the one that is equal to the given element. You may assume that the given element will always be present in the given list lst.
- **is_game_over(money, lst)**
  → If there's no money anymore, or the given list is empty, then this function should return True. Otherwise False.

2) THE ALGORITHM

This part contains the most important function to implement ==> the function **start_game()**.
This function has no parameter. It allows only to launch the game using all the previously defind functions.
But don't worry. I have written a pseudo code for the implementation of this function. Pseudo code means lines of comments you write down explaining what your algorithm should do at every line without being dependent of a specific programming language. You just need to translate it correctly to python. But before translating … first understand how it works!

Only if you have really tried and you're still struggling with start_game(), then you can open LOTTO2.py. That file contains the same as in LOTTO1.py except for the function start_game() that is already implemented.
Remember than there is no universal way of implementing this function. So if your implementation is different than mine, than that's completely normal.

Your time now ! Have fun coding !

Example

Imagine I start the game and enter the name “Adam” and password 1234 (password can also be a string if you prefer).
Then I put 1000 euros in the game.
Then imagine a generated list by the program: [12,15,22,5,9]

First guess: Number 9
→ That number is present in the list, the program prints: *“Congratulations, you now have only 4 numbers to guess, your cash is now 2000 euros”*.
The list now needs to remove the number 9 and becomes: [12,15,22,5]

Guess 2: 1
→ *“Fail ! You have now only 1800 euros left, and you still need to guess 4 numbers.”*
(Because 1/10 of 2000 is 200. So 2000-200 = 1800)
The list remains: [12,15,22,5]

Guess 3: 9
This number has already been guessed and is therefore not present in the list anymore.
→ *“Fail ! You have now only 1620 euros left, and you still need to guess 4 numbers.”*
The list remains: [12,15,22,5]

Guess 4: 97
Even if the number is not between 1 and 30, he loses his guess and 10% of his money is gone.
→ *“Fail ! You have now only 1458 euros left, and you still need to guess 4 numbers.”*
The list remains: [12,15,22,5]

Guess 5: 15
→ *“Congratulations, you now have only 3 numbers to guess, your cash is now 2916 euros”*
The list becomes: [12,22,5]

Etc. Etc. The user can guess multiple times until he has no money or no numbers to guess anymore.

You’ll understand that this game is not a simulation of any realistic guessing machine :p. It’s just for the purpose of learning, we’re not developing any real lotto system.