SE226 – LAB#5 2022-2023 SPRING

Aim: Understanding functions and data structures.

Please download the “1000words.txt” file in BlackBoard under Lab2 folder again. It will give you the list declaration for 1000 words as “words” variable.

1. Remember the work you have done in the previous week. This time, to generate a password, you will write a function named: **passGen**. Write a Python script that will generate random passwords from the given list variable `words`, that you will get by utilizing the array declaration code given to you. Then, your script should assign the variable `user_number` to a value to be entered by the user. Your script should validate that the value of `user_number` must be between 3 and 7, inclusive. This value would refer to the number of words to be chosen randomly from the list variable `words`. You will send this number as a parameter to your passGen function. Depending on the value of `user_number`, your code should randomly choose several words from the `words` list. Lastly, your script should concatenate these words into one string and reverse it to create the password and return that string as a result.

Note: You may use your code from previous week for this task.

2. Define the following two functions in your code:

`rep_with_upper` - This function takes a string as its only parameter and returns a new string value. The function should create the new string value by replacing a character randomly chosen inside the given parameter with its uppercase version (e.g. a --> A). If there are more than one of that character, make sure that all of them are replaced. In your modified script, you will call this function by passing the password created in Question#1 as the argument value.

`swap_letters` - This function takes a string as its only parameter and returns a new string value. The function should create the new string value by swapping the first two characters of the given parameter with the ones in the last two (e.g. Password --> rdsswoPa). In your modified script, you will call this function by passing the password returned from `rep_with_upper` function as the argument value.

Now, modify your script for Question#1 in the following manner: Using your answer to Question#1, create the initial password; then using this password as the argument value to the `rep_with_upper` function create a new password; finally using this new password as the argument value to the `swap_letters` function create the final version of the password. Your modified script should print both the initial and the final versions of the password created.

SAMPLE OUTPUT (bold parts are entered by user):

```
Please enter a value (from 3 to 7) for the number of words: 13

Invalid value!

Please enter a value (from 3 to 7) for the number of words: 1 Invalid

value!

Please enter a value (from 3 to 7) for the number of words: 4

Initial password: xinunomelerusnohtyp

Final version of the password: ypnunOmelerusnOhtxi
```

3. Modify your script for Question#2 by adding the following function: `search_letter`

`search_letter` - This function takes two strings and returns a bool value (i.e., either `True` or `False`). The first argument value will be the source string to be searched and the second argument value will be the key letter to be searched in the source. If the key letter is found in the source the function will return `True`, otherwise `False`.

Test your `search_letter` function in the following way: Call the function by passing the final version of the password created in Question#2 as the first argument value and your name's first letter as the second argument value. Check the function's return value to see the result.

4. Last task: Do the tasks 1, 2 and 3 again in C++.