**Munster Technological University - Kerry**

**Ord/Hons BSc. in Computing with Specialism - Year 1**

**Continuous Assessment #2**

**Date: 11/3/24**

**Time: 9.10 – 10.45 a.m.**

**Structured Programming 2**

**Instructions:** Copy this file to the desktop of your machine. You may also download and refer to the file **PythonSpellChecker.docx** during the CA. Disconnect from the Internet at the start of the CA and close all other Python module materials that you might have access to, before attempting the following question.

Launch IDLE and create an empty Python file called ***FirstnameLastname***.py on the desktop. When you are finished coding, upload your Python file through Canvas. You should keep a copy of your Python file on your OneDrive, unaltered, in case I need to access it at a later stage.

Please don’t logout of Canvas until I have confirmed that I have received your code successfully.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Q1.**

An organisation wishes to enforce a strict password creation model for its workforce. In this model, a password is only valid if it passes the following tests:

* Must contain between 8 and 15 characters inclusive
* Must begin with a symbol and end with a symbol. The symbols allowed in this case are £ % & \* . and #
* The first character cannot be the same as the last character
* Must contain only letters (uppercase and lowercase), digits and values from the following set of symbols £ % & \* . and #
* Must be at least 1 uppercase letter, 1 lowercase letter, 2 digits and 2 symbols in the password
* It will be illegal to have three consecutive characters all with the same values e.g. the substring “777” or “AAA” would be illegal, but the substring “AaA” would be legal – note that you will need a **loop** for this part

Your program should read in an arbitrary number of password values (use **return** i.e. empty string as the sentinel value) and validate each one according to the criteria outlined above. You should perform the tests **in the order indicated above**, starting in each case by testing the length of the strings entered. As long as the value entered remains invalid, it should be rejected with the user given the reason for the **first point of failure**, along with a prompt to re-enter.

Once the main loop has finished, the program should simply display a message-box indicating the list of valid passwords that have been entered. Should no valid passwords be entered by the time the main loop completes, then a message-box will appear to indicate this.

For full marks here your program should, along with a logically correct solution for the problem above, include a **comment** at the top of the program which should briefly explain the purpose of the program.

Your program should run as indicated in the following sample screenshots. Also, you can use the test values indicated when testing your own program.

**Sample Screenshots**

**Run 1:**

**In this case, the user enters nothing for the password and so the main loop exits immediately. A message-box is displayed to indicate that no valid passwords have been entered.**

A screenshot of a computer error

Description automatically generated A screenshot of a computer error

Description automatically generated

**Run 2:**

**Here the user begins by entering several invalid password values. The user is told what is wrong with the value supplied each time (the first point of failure) and the validation loop only quits once a valid password value has been supplied.**

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error message

Description automatically generated

A screenshot of a computer error message

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

**At last a valid password value is supplied so the validation loop exits and the user gets asked for another password value – this, and subsequent values, will then be validated in the same way as before**

A screenshot of a computer error

Description automatically generated

**When the user hits return, the main loop exits and a message box appears showing the list of valid stock ID values that have been entered.**

A screenshot of a computer

Description automatically generated