TURIN POLYTECHNIC UNIVERSITY IN TASHKENT

COURSE OF COMPUTER SCIENCE
LABORATORY PRACTICE n. 6

Exercise 1:

Take the following Python code that stores a string:

```
str = 'X-DSPAM-Confidence:0.8475'
```

Use `find` and string slicing to extract the portion of the string after the colon character, and then use the float function to convert the extracted string into a floating point number.

Exercise 2:

Write a while loop that starts at the last character in the string and works its way back to the first character in the string, printing each letter on a separate line.

Sample Output:

```
Enter a string: hello
o
l
l
e
h
```

Exercise 3:

Write down a Python program which:

- reads a string without spaces from the keyboard (with length at most equal to 50 characters).
- checks whether it is palindrome, displaying a proper message on screen based on the test result.

Notice: capital letters should be considered as equivalent to their corresponding small version!
Recall that a string is said to be palindrome when it can be read equivalently, from left to right and from right to left. For instance, strings “Anna”, “83238” and “AbCcBa” are palindrome.

Exercise 4:

Let’s say, we are given a string that contains a combination of the lower and upper case letters. Write a program to arrange the characters of a string so that all lowercase letters should come first.

Sample Output:

```
Enter a string: AbcDfGhJKlm
Result: bcfhlmADGJK
```

Exercise 5:

Write down a Python program able to:

- read two words `s1` and `s2`, each one consisting of at most 20 characters.
- generate and display a new string `s3` by removing from `s1` all characters appearing in `s2` (capital letters should be considered as different from the corresponding small ones).

Example: The following is a possible execution example (underlined text is typed by the user):

```
Input the s1: Example
Input the s2: exam
Resulting s3: Epl
```

Exercise 6:

Write down a Python program in order to:

- read a single character `ctr`.
- read a sequence of words, each of which with length at most equal to 20 characters. The reading operation terminates when the word "`stop`" is introduced.
- display on screen the word in which `ctr` appears most frequently.

Example: The following is a possible program execution (underlined text is typed by the user):

```
Input character: e
Input word: yellow
Input word: green
Input word: orange
Input word: blue
Input word: black
Input word: stop
The word with most 'e' is "green".
```

Exercise 7:

Write a program able to:

- read from the keyboard an unknown number of characters (at most 80), all specified into a single line (terminated by a new line character).
- print on the sub-sequent line the same sequence of characters, where the first character of every "word" has been made uppercase and the remaining ones lowercase.

It is illegal to read and process one word a time: the row introduced by the user must be entirely stored into an array and then properly processed.

Example: The following is a possible execution example (underlined text is typed by the user):

```
Input line: the DEVIL hides in the DETAILS.
Output line: The Devil Hides In The Details.
```

Exercise 8:

Write down a Python program which:

- reads three strings `s1`, `s2` and `s3` from the keyboard (each one with length at most equal to 50 characters).
- generates a new string by replacing all the occurrences of `s2` within `s1` with `s3`.

- outputs such a resulting string.

Example: The following is a possible execution example (underlined text is typed by the user):

```
Input 1st string: abcde12345cdefg
Input 2nd string: cde
Input 3rd string: #####
Resulting string: ab#####12345#####fg
```

Exercise 9:

Develop a program that counts the number of letters, digits, and special symbols from a given string.

Sample Output:

```
Enter a string: P@#yn26at^&i5ve
Letters: 8
Digits: 3
Symbols: 4
```

Exercise 10:

Develop a program that calculates the sum and average of the digits existing in a given string.

Sample Output:

```
Enter a string: P@#yn66at^&i5ve
Sum: 15
Average: 5.6666666667
```