IF100
Functions - Practice #1

Introduction

The aim of this example is to practice functions and loops in Python. The use of loops is due to the nature of the problem; that is, you cannot finish this practicing assignment without using loops. On the other hand, although you can solve the problem described below without using functions, we highly recommend the use of functions, so that you can practice on this topic.

Description

In this exercise, you will implement a simple calculator that performs a set of operations with at most two operands.

Your program will input an arithmetic operation that contains an operator symbol and two operands. The operands should be _non-negative_ integers and the operators can be one of the following:

Operator	Arithmetic	Example
+	addition	4+5
-	subtraction	2-8
*	multiplication	45*2
/	division	8/2

You program will first check the validity of the input arithmetic operation and continuously prompt for another input arithmetic operation until the user enters a valid one. A valid arithmetic operation (i) should contain exactly 1 operator from the set of operators mentioned above, (ii) should not have the operator in the beginning or in the end, (iii) should contain at least 2 operands of at least 1 digit, and (iv) should not have any other symbols.

When a valid input arithmetic operation is entered by the user, your program will perform the required arithmetic operation, meaning that it will evaluate the valid input arithmetic operation, and display the result of this operation.

After evaluating the arithmetic operation and displaying the respective result, your program should re-prompt for another arithmetic operation and follow the

steps described above. This repetition should continue until the user enters the exact phrase of "stop".

Please see "Sample Runs" section in order to understand the flow of the program, the inputs and outputs in a better way.

We highly recommend the use of functions. It would be easier if you first apply decomposition and then implement the solutions of the subproblems in different functions.

Hint #1: You may write a function that takes the arithmetic operation (like *"3+15"* or *"3a+f4"*) as its only parameter; returns true if the arithmetic operation is valid, and false otherwise. Alternatively, you can write a number of different functions to check a specific characteristic that the arithmetic operation should have.

Hint #2: You may write a function that takes the arithmetic operation (like *"3+15"*) as its only parameter and parses the input arithmetic operation into its operands and the operator.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user.

Sample runs are not %100 comprehensive. You are required to read the whole documentation and decide on what other cases you might try your program with.

Sample Run

Please enter an arithmetic operation: ***abc***
Invalid arithmetic operation!

Please enter an arithmetic operation: ***345***
Invalid arithmetic operation!

Please enter an arithmetic operation: **+**
Invalid arithmetic operation!

Please enter an arithmetic operation: ***128-***
Invalid arithmetic operation!

Please enter an arithmetic operation: ****12***
Invalid arithmetic operation!

Please enter an arithmetic operation: *12%2*
Invalid arithmetic operation!

Please enter an arithmetic operation: *33/x*
Invalid arithmetic operation!

Please enter an arithmetic operation: *3+5*
3+5 = 8

Please enter an arithmetic operation: *Stop*
Invalid arithmetic operation!

Please enter an arithmetic operation: *STOP*
Invalid arithmetic operation!

Please enter an arithmetic operation: *83*121*
83*121 = 10043

Please enter an arithmetic operation: *stop*
Bye...