**IF100**

**Practice 5**

**Introduction**

The aim of this example is to practice the sequences in Python.

**Description**

In this exercise, you will write a Python program that will enable the user to retrieve an element from a predefined list. You can reach the list that you will be using in ***givenList.ipynb*** (in the same folder with this document). In this file (*givenList.ipynb*), there is a variable named *myList* and the program that you will write will make use of this particular sequence of integers.

This exercise is intended to be solved part by part, as described below:

Part-1

In this part, your program will start with prompting for the index of the element to be retrieved. Then, it will check the validity of this value. Since the indices of sequences start from 0, the value obtained from the user should be greater than or equal to 0, and since the last index of a sequence is one less than the number of elements of that sequence, the value obtained from the user should be smaller than the number of elements of the given list. If the user enters an invalid value for the corresponding index, then your program will display an error message. Otherwise, if the user enters a valid value, then your program will simply display the element that sits at the given position.

Part-2

In this part, you will extend the program that you have implemented in the first part, so that it will enable the user to retrieve a sublist from the given predefined list. Hence, in this part, your program will prompt for two inputs: the starting index and the stopping index of the sublist to be retrieved. Your program should check the validity of both of these inputs one by one. Whenever the user enters an invalid value, your program should display an appropriate error message and stop its execution. The validity check for the starting index is the same as it was in the first part; but in the validity check for the stopping index, you should also consider the fact that the stopping index must be greater than the starting index. In case that both of the inputs are valid, then your program should display the part of the list that starts at the starting index (included) and ends in the stopping index (excluded).   
  
Note that the stopping index will be inputted only if the starting index is valid. If starting index is not valid, then do not input for the stopping value and you should terminate your program at this point with an appropriate message.

Part-3

In this part, you will extend the program that you have implemented in the second part, so that it will enable the user to retrieve a part (some elements in an order) from the given predefined list. This time, your program will prompt for three inputs: the starting index, the stopping index and the step size. Your program should check the validity of all of these inputs one by one. Whenever the user enters an invalid value, your program should display an appropriate error message and stop its execution. The validity check for the starting index and the stopping index are the same as they were in the second part. On the other hand, the value of the step size has to be a positive integer smaller than or equal to the difference between the starting index and the stopping index.

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

*You should not make any assumptions on the number of elements of the utilized list; the only thing you know is indeed there will be a list of integers named as myList.*

**Input and Output**

The inputs of the program and their order are explained below. Please see "Sample Runs" section for some examples.

There are three inputs to your program, in the order given below:

1. Starting index (of type int)
2. Stopping index (of type int)
3. Step size (of type int)

The inputs should satisfy the following six rules:

1. Starting index must be greater than or equal to 0
2. Starting index must be smaller than the number of elements of the given list
3. Stopping index must be strictly greater than the starting index
4. Stopping index must be smaller than the number of elements of the given list
5. Step size must be strictly greater than 0
6. Step size must be smaller than or equal to the difference between the starting index and the stopping index

The prompts of the input statements to be used has to be exactly the same as the prompts of the "Sample Runs".

If all the inputs are valid, then there will only be one output of your program: the part of the list that sits in the given positions. The format of this output must be exactly as follows:

*The values between indices starting-index* and *stopping-index   
that are step-size apart from each other are list-part.*

If there is an invalid input, then again there will only be one output of your program: the underlying reason of error. In that case, the output can be one of the following:

* *Invalid starting value!*
* *Invalid stopping value!*
* *Invalid step size!*

**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 Run 1**

Please enter the starting index: ***-1***

Invalid starting value!

**Sample Run 2**

Please enter the starting index: ***33***

Please enter the stopping index: ***48***

Please enter the step size: ***0***

Invalid step size!

**Sample Run 3**

Please enter the starting index: ***140***

Please enter the stopping index: ***149***

Please enter the step size: ***10***

Invalid step size!

**Sample Run 4**

Please enter the starting index: ***148***

Please enter the stopping index: ***149***

Please enter the step size: ***1***

The values between indices 148 and 149 that are 1 apart from each other are [16].

**Sample Run 5**

Please enter the starting index: ***140***

Please enter the stopping index: ***149***

Please enter the step size: ***9***

The values between indices 140 and 149 that are 9 apart from each other are [48].

**Sample Run 6**

Please enter the starting index: ***0***

Please enter the stopping index: ***10***

Please enter the step size: ***2***

The values between indices 0 and 10 that are 2 apart from each other are [16, 10, 15, 47, 47].