**IF100**

**Practice 3**

**Introduction**

The aim of this example is to practice the conditional if statements. The use of if statement is due to the nature of the problem; that is, you cannot finish this exercise without using if statements.   
 **Description**

In this exercise, you will write a Python program that prompts the user to enter a few pieces of information regarding a student's records and find out whether the student can graduate or not. Specifically, your program will get the gathered SU Credits, gathered ECTS credits, whether (s)he has taken SPS303, and the current GPA of the student, respectively. You may assume that the types of the inputs are always correct (i.e. user will always enter integer values for SU Credits and ECTS Credits; and will always enter a float value for the GPA).

However, we have some possible values for each input. For instance, the input value for the SU Credits credits must be between 0 and 200, both inclusive. Therefore, your program needs to check whether this value is in range [0,200]. If the user enters an invalid value for the SU credits, then your program will display an error message and stop the execution, meaning that it won’t continue with the remaining operations. Otherwise, if the user enters a value in the correct range for the SU credits, your program will continue and ask about the ECTS credits. It should be between 0 and 400, both inclusive. If this input does not fall into the valid range (not necessarily sufficient), your program should again display an error message and stop the execution i.e, it will not continue with any other operations. The same procedure applies for SPS303 passing situation where valid values can only be "Yes" or "No" and current GPA where valid range is [0, 4.0] .

After being able to collect all the inputs from the user, your program will decide whether the graduation requirements are met. These requirements are given in the following sections of this document.

**Inputs and Outputs**

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

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

1. SU Credits (of type integer)
2. ECTS Credits (of type integer)
3. SPS303 passing situation (of type string)
4. GPA (of type float)

The prompts of the input statements have to be exactly the same as the prompts of the "Sample Runs". Additionally, you may assume that the user will enter valid types for the inputs.

Your program should perform input check for its inputs as listed below:

* SU Credits should be between 0 and 200, both inclusive
* ECTS Credits should be between 0 and 400, both inclusive
* SPS303 passing situation should be either "Yes" or "No" (case-sensitive!)
* GPA should be between 0 and 4.0, both inclusive

If the user enters an erroneous input, then your program should display an appropriate error message and it should not ask for the latter inputs.

If all the inputs are valid, then there will only be one output of your program that is the result of the graduation situation. The format of this output must be exactly as one of the following:

*This student can graduate.*

*This student cannot graduate.*

The decision is based on the following criteria of graduation:

* Minimum 125 SU Credits
* Minimum 256 ECTS Credits
* SPS303 should have been passed by the student
* Minimum GPA of 2.00

All of these criterias should be satisfied together to be able to graduate. For instance, having only minimum 125 SU Credits is not enough itself for graduation.

If there is an invalid input, then again there will only be one output of your program: the reason of not being able to carry out the requested operation. In that case, the output should be one of the following depending on the error:

* *Incorrect SU Credits!*
* *Incorrect ECTS Credits!*
* *Incorrect SPS303 passing situation!*
* *Incorrect GPA!*

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

**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. The introductory line is mandatory as well. You may not change the introductory line or any of the prompts sentences. Your program should be presented exactly like these sample runs.

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 1 (***Invalid sps303 input value***)**

Welcome to graduation estimator. Let us serve you!  
Please enter the SU Credits for the student: ***200***  
Please enter the ECTS Credits for the student: ***138***  
Please enter the SPS303 passing situation for the student: ***no***  
Incorrect SPS303 passing situation!

**Sample Run 2 (***Invalid su credits input value***)**

Welcome to graduation estimator. Let us serve you!  
Please enter the SU Credits for the student: ***300***  
Incorrect SU Credits!

**Sample Run 3 (***Invalid gpa input value***)**

Welcome to graduation estimator. Let us serve you!  
Please enter the SU Credits for the student: ***30***  
Please enter the ECTS Credits for the student: ***40***  
Please enter the SPS303 passing situation for the student: ***Yes***  
Please enter the GPA for the student: ***4.20***  
Incorrect GPA!

**Sample Run 4 (***All inputs are valid***)**

Welcome to graduation estimator. Let us serve you!  
Please enter the SU Credits for the student: ***125***  
Please enter the ECTS Credits for the student: ***262***  
Please enter the SPS303 passing situation for the student: ***Yes***  
Please enter the GPA for the student: ***2.00***  
This student can graduate.

**Sample Run 5 (***All inputs are valid***)**

Welcome to graduation estimator. Let us serve you!  
Please enter the SU Credits for the student: ***126***  
Please enter the ECTS Credits for the student: ***252***  
Please enter the SPS303 passing situation for the student: ***Yes***  
Please enter the GPA for the student: ***1.19***  
This student cannot graduate.

**Sample Run 6 (***Invalid ects credits input value***)**

Welcome to graduation estimator. Let us serve you!  
Please enter the SU Credits for the student: ***50***  
Please enter the ECTS Credits for the student: ***-1***  
Incorrect ECTS Credits!

**Programming and Coding Advice**

It would be easier for you to implement the algorithm of this problem if you first try to draw the flowchart or write the pseudocode so that you can go over your solution to see if there are any errors.

Additionally, you can decompose the coding (and algorithm design) process into smaller pieces: it is always a good idea to divide the problem into smaller and easier subproblems and to conquer them one by one to get the overall solution of the problem.