Introduction

The aim of this take-home exam is to practice decision making (conditional if statements), sequences, and methods. The use of if statements and string methods are due to the nature of the problem; that is, you cannot finish this take-home exam without using them.

Description

In distance learning, a university added Satisfactory(S)/Unsatisfactory(U) options to their grading system and asked you to write a Python program that takes the following three (3) information as inputs, regarding a particular student:

1. Names of the courses that were taken by the student in the previous semester(s), together with their corresponding letter grades, in a predetermined format
2. Names of the courses that are being taken by the same student in the current semester, together with their corresponding letter grades, in a predetermined format
3. Name of the course, for which the grade will be checked

By using the course name given by the student and the letter grade information, your program then should calculate if the student will be able to choose between S and the letter grade, or if the student's grade will be U or F.

Originally, letter grade options for courses are; A, A-, B+, B, B-, C+, C, C-, D+, D, and F, where F is the failing grade. However, the university wanted to give their students the option to prevent their GPAs from being affected in a negative way due to the pandemic conditions by choosing the Satisfactory(S) option if they manage to get a passing grade; in other words, a grade other than F.

Students can take a course multiple times. In this respect, the letter grade options that the students will receive differ according to both the previous grade and the current grade of a particular course. Please follow the items listed below, or the flowchart given on the next page, for further details.

- If the current grade of a course is something other than F, students can keep their letter grade or they can choose the S grade.
- If the current grade of a course is F, *and*
  - if the student is taking this course for the first time, then the student's final grade will be U, *or*
  - if the student has taken this course before, then your program should check the previous course grade of this student.
    - If the previous grade is U, then the student's final grade should remain as U.
    - Otherwise, the student's final grade will be F.
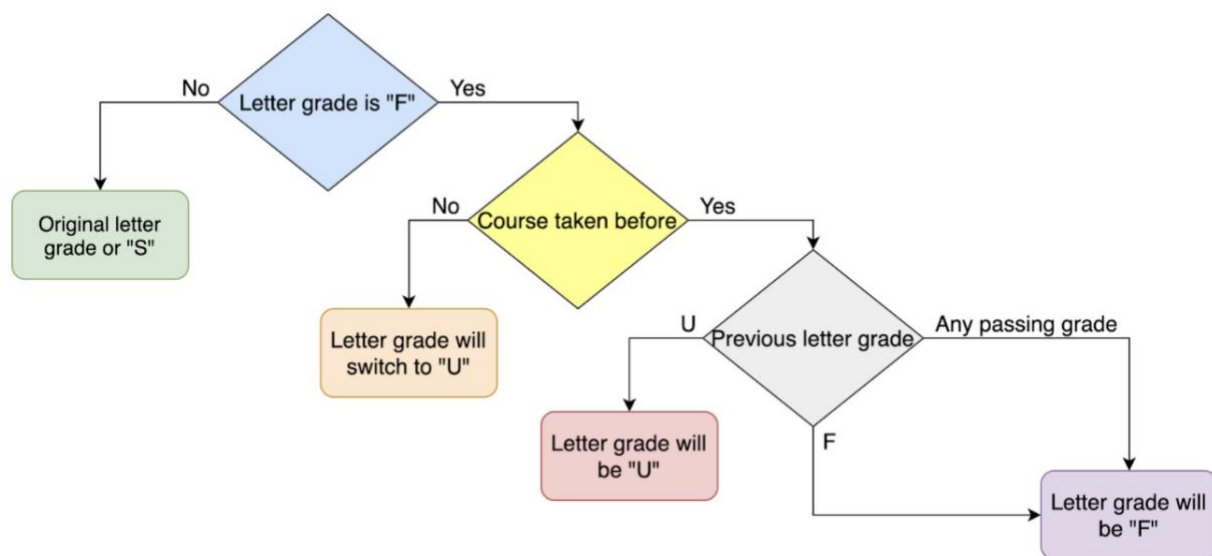
Inputs, Input Checks

The inputs of the program and their order are explained below. It is extremely important to follow this order with the same characters since we automatically process your programs. **Thus, your work will be graded as 0 unless the order is entirely correct**. Please see the "Sample Runs" section for some examples.

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

Here is the detailed information on the inputs and the input checks:

- Courses that were taken in the previous semester(s) - *previous courses list*
  <u>*Format*</u>*: course1:grade1;course2:grade2;...;courseN:gradeN*
     i.e. **"MATH101:A-;SPS101:b;IF100:B+;HIST191:s"**

  - Colon (":") is used between each one of the course names and letter grades. First comes the course name, and after the colon, there comes the letter grade. The length of the course name can be anything. Therefore, you cannot make any assumption on the course name length. For example, the course name can be "MATH101" or "IF100" or "CS5008", whose lengths are 7, 5, and 6, respectively.
  - Semicolons (";") are used between course name and letter grade pairs.
  - The user may enter extra colons (":") and/or semicolons (";") within the input mistakenly. Thus, your program should check if the number of colons is 1 (one) more than the number of semicolons. If not, your program should display an error message and terminate. You may assume if the input is given with additional colon and/or semicolon characters, then "colon count = semicolon count + 1" equality will be broken.
  - Letter grades are case-insensitive, which means that some of the letter grades could be entered in upper-case, whereas some of them could be entered in lower-case. However, you may <u>assume</u> that the users will always enter a valid letter grade. That is, the user will not enter a letter grade except for A, A-, B+, B, B-, C+, C, C-, D+, D, F, S, U and their lowercase versions.
  - You may also <u>assume</u> that the course names will consist only of capital letters and digits.
  - The number of course name and letter grade pairs is not fixed, and there is no predefined or pre-assumed order in this input, either with respect to the letter grades or to the course names.
  - You may <u>assume</u> that there won't be any duplicated course names, no spaces, or any other characters within this input.
  - You may check *Sample Runs,* for more examples. However, please ***keep in mind*** that sample runs may not cover all possible cases mentioned in   this   document.

- Courses that are being taken in the current semester - *current courses list*
  <u>*Format*</u>*: course1:grade1;course2:grade1;...;courseN:gradeN*
     i.e. **"MATH102:a-;SPS102:B;CS201:b+;HIST192:a"**

- ○ Rules and details are the same as above.
- ○ Again, you may check *Sample Runs* for better understanding.

- ● Course name to be searched - *course name*
  Format: `coursename`

  - ○ To calculate the final grade or grade options, the student should be currently enrolled in the course given as `coursename`. Thus, your program should initially check if `coursename` exists in *the current course list*. If not, the program should print an appropriate error message and terminate.
  - ○ If `coursename` exists in the *current course list*, then your program should continue as explained in the description part. You may check the flowchart above for a better understanding.
  - ○ You may <u>assume</u> that all course names will consist only of capital letters and numeric code in string format.

Output

If the user enters an invalid input, then your program should display an error message saying "Invalid input", and if the `coursename` does not exist in the `current course list,` then your program should display an error message saying "You didn't take `coursename` this semester." In any erroneous case, your program should terminate after displaying the message and <u>*without taking any further inputs*</u> or <u>*without displaying any other results*</u>.

If the student's grade to be displayed is F or U, then the format of the output will be "Your grade for *coursename* is F." or "Your grade for *coursename* is U.".

If the student's current grade is anything other than F, then the format of the output will be "You can choose between S and `current_grade` for *coursename*.".

You may check the "Sample Runs" section given below for some examples.

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. <u>You have to display the required information in the same order and with the same words and characters as below</u>.

Sample Run 1
```
Please enter the courses you have taken previously with letter grades:
```
MATH101:a;SPS101:B;CS201:B+;HIST191:D;CS204:f;;CS210:S
```
Invalid input
```

Sample Run 2
```
Please enter the courses you have taken previously with letter grades:
```
MATH101:A;SPS101:B;CS201:B+;HIST191:D;CS204:F;CS210:S:
```
Invalid input
```

Sample Run 3
```
Please enter the courses you have taken previously with letter grades:
```
MATH101:A;SPS101:B;CS201:B+;HIST191:D;CS204:F;CS210:S
```
Please enter the courses you have taken this semester with letter grades:
```
CS210:F;CS206:F;MATH203:C;HIST191:F;CS204::F
```
Invalid input
```

Sample Run 4
```
Please enter the courses you have taken previously with letter grades:
```
MATH101:A;SPS101:B;CS201:B+;HIST191:D;CS204:F;CS210:S
```
Please enter the courses you have taken this semester with letter grades:
```
CS210:F;;CS206:F;MATH203:C;HIST191:F;CS204:F
```
Invalid input
```

Sample Run 5
```
Please enter the courses you have taken previously with letter grades:
```
MATH203:C;HIST192:A;CS201:A-;CS210:C
```
Please enter the courses you have taken this semester with letter grades:
```
MATH203:C;HIST192:A;CS201:A-;CS210:C
```
Please enter the course you want to check: ***CS408***
You didn't take CS408 this semester.
```

Sample Run 6
```
Please enter the courses you have taken previously with letter grades:
```
MATH101:a;SPS101:B;CS201:B+;HIST191:D;CS204:F;CS210:c-

Please enter the courses you have taken this semester with letter grades:
CS210:a-;CS206:F;MATH203:a;HIST191:F;CS204:f
Please enter the course you want to check: *CS204*
Your grade for CS204 is F.

Sample Run 7
Please enter the courses you have taken previously with letter grades:
MATH101:A;SPS101:b;CS201:B+;HIST191:D;CS204:F;CS210:u
Please enter the courses you have taken this semester with letter grades:
CS210:F;CS306:b+;MATH203:C;HIST192:F;CS204:b
Please enter the course you want to check: *CS210*
Your grade for CS210 is U.

Sample Run 8
Please enter the courses you have taken previously with letter grades:
MATH101:A;SPS101:B;CS201:b+;HIST191:D;CS204:F;CS210:A-
Please enter the courses you have taken this semester with letter grades:
CS306:F;MATH203:C;HIST192:c-;CS201:F
Please enter the course you want to check: *CS201*
Your grade for CS201 is F.

Sample Run 9
Please enter the courses you have taken previously with letter grades:
MATH101:A;SPS101:a;CS201:B+;HIST191:D;CS204:B-;CS210:b+
Please enter the courses you have taken this semester with letter grades:
CS306:F;MATH203:C;HIST192:A;CS201:A-
Please enter the course you want to check: *CS306*
Your grade for CS306 is U.

Sample Run 10
Please enter the courses you have taken previously with letter grades:
MATH101:A;SPS101:B;CS201:b+;HIST191:D;CS204:B-;CS210:a-
Please enter the courses you have taken this semester with letter grades:
MATH203:d+;HIST192:A;CS201:A-;CS210:c-
Please enter the course you want to check: *CS210*
You can choose between S and C- for CS210.

Sample Run 11
Please enter the courses you have taken previously with letter grades:
MATH101:A;SPS101:F;CS201:b+;HIST191:D;CS204:B-;CS210:a-

```
Please enter the courses you have taken this semester with letter grades:
SPS101:A;MATH203:d+;HIST192:A;CS201:A-;CS210:c-;PS101:F
Please enter the course you want to check: PS101
Your grade for PS101 is U.
```

How to get help?

You can use GradeChecker (https://learnt.sabanciuniv.edu/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

What and where to submit?

You should prepare (or at least test) your program using Python 3.x.x. We will use Python 3.x.x while testing your take-home exam.

It'd be a good idea to write your name and lastname in the program (as a comment line of course). <u>Do not use any Turkish characters anywhere in your code (not even in comment parts).</u> If your name and last name is "İnanç Arın", and if you want to write it as comment; then you must type it as follows:
        # Inanc Arin

Submission guidelines are below. Since the grading process will be automatic, students are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be 0.

- Download your code as *py* file with "File" -> "*Download .py*" as below:
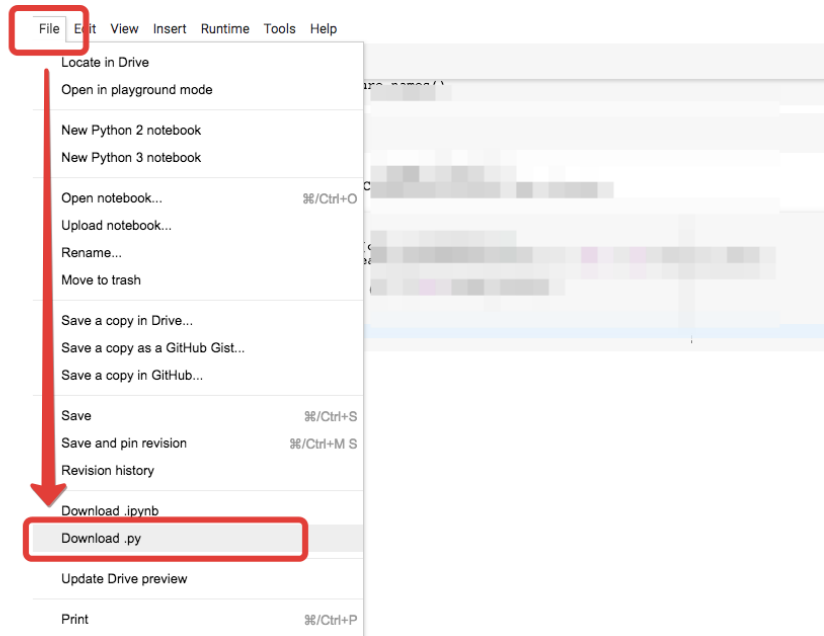
- Name your *py* file that contains your program as follows:

    "username_the2.py"

    For example: if your SUCourse+ username is **"duygukaltop"**, then the name of the *py* file should be: **duygukaltop_the2.***py* (please only use lowercase letters).

- Please make sure that this file is the latest version of your take-home exam program.

- Submit your work **through SUCourse+ only**! You can use the GradeChecker only to see if your program can produce the correct outputs both in the correct order and in the correct format. It will not be considered as the official submission. You must submit your work to SUCourse+.

- If you would like to resubmit your work, you should first remove the existing file(s). This step is very important. If you do not delete the old file(s), we will receive both files and the old one may be graded.

General Take-Home Exam Rules

- Successful submission is one of the requirements of the take-home exam. If for some reason, you cannot successfully submit your take-home exam and we cannot grade it, your grade will be 0.
- There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse+ time and your computer time <u>may</u> have 1-2 minute(s) differences. You need to take this time difference into consideration.
- Do NOT submit your take-home exam via email or in hardcopy! SUCourse+ is the only way that you can submit your take-home exam.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please do submit your **<u>own</u>** work only. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the syllabus of the course.

Good luck!
Elif Pınar Ön, Ethem Tunal Hamzaoğlu
       & IF100 Instructors