IF100 – Spring 2019-2020
Take-Home Exam #3
Due December 11nd, Friday, 23:55 (Sharp Deadline)

Introduction

The aim of this take-home exam is to practice on loops (for/while statements). The use of loops is due to the nature of the problem; that is, you cannot finish this take-home exam without using for/while statements.

Description

CineSU movie theater asked you to write a program to take reservations from users for the screening movies. If there are not enough seats in the movie that the user wants to watch, then the program should suggest alternative movie(s) from the same genre, which have enough seats for reservation.

In this take-home exam, you will implement a Python program that will get three inputs from the user.

- The first input of your program is the movies' information: movie names, movies' genres, and the remaining seats of these movies, written in a predefined format. Here, you should keep in mind that a movie might occur several times in this input, in the case of which your program should sum up all the remaining seat counts declared associated with the same movie name. In this input, information of movies will be separated with commas.
- The second input of your program is the name of the movie that the user wants to watch.
- The third input is the number of people to make reservations for.

You need to print an appropriate output according to some conditions in your program.

- If the movie is not in the theaters (i.e. does not exist in the first input), then your program needs to print out a message accordingly.
- If the movie is in the theaters and there are enough seats, then your program should print a confirmation message about the reservation.
- If the movie is in theaters, but there are not enough seats left, then your program needs to suggest other movie(s) from the same genre that have enough seats for reservation.

You can find the details about the inputs and outputs in the following section.

Input, Process and Output

The inputs of the program and their order are explained below. It is extremely important to follow this order with the same format since we automatically process your programs. Also, prompts of the input statements to be used have to be exactly the same as the prompts of the "Sample Runs". ***Thus, your work will be graded as 0 unless the order is entirely correct***.

Your program will have multiple inputs and the specifications for these inputs are explained below.

- First input includes movies' information; names of the movies in the theater, number of available seats (quota) and genre in the following format:

  movie#1:movie#1_quota:movie#1_genre,movie#2:movie#2_quota:movie#2_genre…,movie#N:movie#N_quota:movie#N_genre

  - You may assume that each movie's name will be given correctly and there will be no comma ("**,**") and colon ("**:**") characters used in the movie name or the movie genre. Keep in mind that the same movie name might occur more than once in the given input.
  - You may assume that even if the movie name occurs more than once, the genre given with that movie name will be the same. There will be only one genre for a particular movie.
  - You may assume that there will be only one comma ("**,**") between eachmovie information and only one colon ("**:**") between the movie name, number of available seats left in the movie and the genre information of the movie. Also, you may assume that the characters in this input will be given in all lowercase.
  - There will not be any comma or colon characters in the beginning or ending of the input.
  - You don't need to perform any input check for this input.

- Second and third inputs will be the name of the movie that wanted to be watched and the person count to make reservations for, respectively.

  - You may assume that the movie name input will be given in all lowercase letters and person count will be given as a positive integer.
  - You don't need to perform any input check for these inputs as well.

Once your program gets the inputs, it should check if the movie name input occurs in the movie information input and if not, it should output;

"There is no such movie in the theater."

If the movie name occurs in movie information input, then your program should check if the total number of available seats in the movie is enough for person count who want to make a reservation to that movie. In order to make this decision, your program should calculate the total number of available seats for the movie by adding up all the number of available seats declared for this movie name in the movie information input. If there is enough available seats, then your program should output;

"The reservation is done!"

If not, i.e. there is not enough available seats for the requested reservation, then your program should determine **other movies from the same genre**, check their total number of of available seats and select the movies that have enough available seats for the person count who wants to make reservations and print names of the movies *in alphabetic order* in the following format. In that case, the output of your program should be;

"There are not enough seats for *requested_movie*! **But you can watch one of the following movies from the genre** movie_genre**:**
*** movie#1**
*** movie#2**
*** ..."**

If there are not enough seats in the movie that the user wanted to make reservations for, and at the same time, there are not enough seats at any other movie from the same genre, then your program should output;

"There are not enough seats for *requested_movie* **and any other movie with the genre** movie_genre**!"**

Please see the "Sample Runs" section 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. You have to display the required information in the same order and with the same words and characters as below.

Sample Run 1

```
Please enter movie names and remaining quota: pulp fiction:0:crime,natural born killers:3:action,reservoir dogs:4:crime
Please enter the movie you want to watch: django unchained
There is no such movie in the theater.
```

Sample Run 2

```
Please enter movie names and remaining quota: pulp fiction:0:crime,natural born killers:3:action,reservoir dogs:4:crime
Please enter the movie you want to watch: dogs
There is no such movie in the theater.
```

Sample Run 3

```
Please enter movie names and remaining quota: pulp fiction:0:crime,natural born killers:3:action,reservoir dogs:4:crime,django unchained:3:drama
Please enter the movie you want to watch: reservoir dogs
Please enter the number of tickets you want to buy: 2
The reservation is done!
```

Sample Run 4

```
Please enter movie names and remaining quota: pulp fiction:2:crime,reservoir dogs:4:crime,django unchained:3:drama,pulp fiction:2:crime
Please enter the movie you want to watch: pulp fiction
Please enter the number of tickets you want to buy: 4
The reservation is done!
```

Sample Run 5

```
Please enter movie names and remaining quota: pulp
fiction:6:crime,reservoir dogs:4:crime,django
unchained:3:drama,natural born killers:3:action,django
unchained:1:drama
Please enter the movie you want to watch: natural born killers
Please enter the number of tickets you want to buy: 4
There are not enough seats for natural born killers and any other
movie with the genre action!
```

Sample Run 6

```
Please enter movie names and remaining quota: pulp
fiction:3:crime,natural born killers:3:action,reservoir dogs:4:crime
Please enter the movie you want to watch: reservoir dogs
Please enter the number of tickets you want to buy: 5
There are not enough seats for reservoir dogs and any other movie with
the genre crime!
```

Sample Run 7

```
Please enter movie names and remaining quota: reservoir
dogs:2:crime,pulp fiction:1:crime,natural born killers:3:action,pulp
fiction:2:crime,reservoir dogs:2:crime,the hateful eight:2:crime,pulp
fiction:1:crime
Please enter the movie you want to watch: the hateful eight
Please enter the number of tickets you want to buy: 3
There are not enough seats for the hateful eight! But you can watch
one of the following movies from the genre crime:
* pulp fiction
* reservoir dogs
```

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.

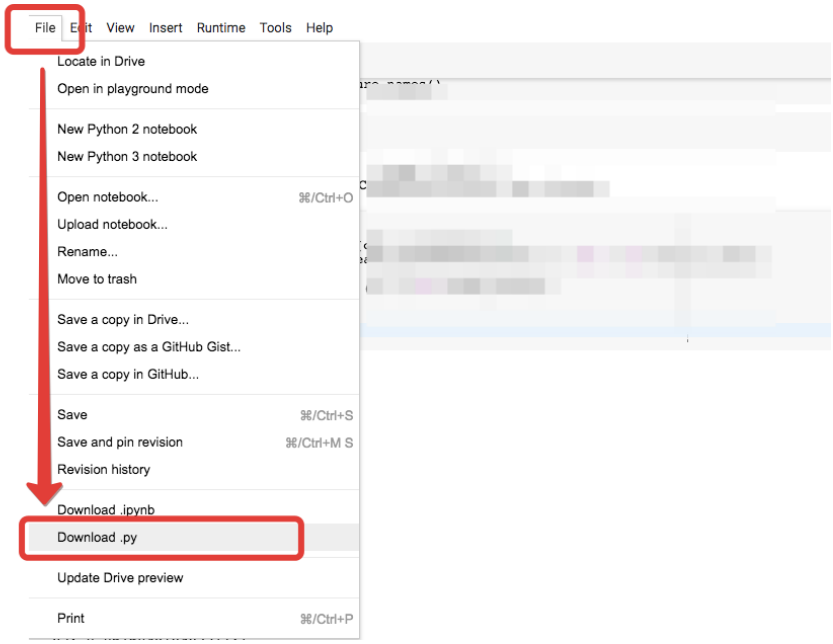
What and where to submit?

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). Do not use any Turkish characters anywhere in your code (not even in comment parts). 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_the3.py"

  For example: if your SUCourse+ username is **"duygukaltop"**, then the name of the *py* file should be: **duygukaltop_the3*.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 may 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 **own** 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