The University of Hong Kong

COMP1117B Computer Programming

Assignment 4 – Difficult version (Full mark 100%)

Due date: April 29, 2022 23:59

Important information

Assignment 4 has two versions, i.e., the easy version (Full marks 70/100) and the difficult version (Full marks 100/100). You should select **one** among the two versions to work on. This document belongs to the **difficult version**.

If you submit answers of both versions, the marks of the difficult version will be used as the final marks of Assignment 4.

Overview

In this assignment, you are going to build a table booking system for a restaurant. Please read this assignment sheet carefully before starting your work.

Reminders

This assignment involves a lot of console input/output. You are reminded that the VPL system on HKU Moodle evaluates your program with a full score under the condition that your **program output is the EXACT MATCH of the expected output**. In other words, any additional or missing space character, newline character, etc., will be treated as errors during the evaluation of your program. Also, you are suggested to **make more test cases** on your own for testing your program.

Assignment Specification and Requirement

The table booking system allows table bookings, table allocations and relevant status checking. The customer who requests a booking needs to provide their personal information, dining date and the number of seats required. Once a booking is added to the system, the booking is in Pending state. The manager of the restaurant can then allocate tables to the booking to confirm the reservation.

To facilitate checking and identification of booking, the system uses a ticket system to identify each booking. For each day, a ticket code sequence is maintained. The ticket code starts from 1 and is incremented by 1 when a new booking comes.

Load tables information from a file

The table information is defined in a text (.txt) file, in which each line defines a table, including the table code and the size (number of seats) of the table. Upon running the program, the program should ask the user to input the name of the text file and then loads the data into the system. Please refer to the sample input file and sample program output below for the exact input and output format.

The text file has the following format:

- Each line defines a table.

- The first field is a two-digits number defining the table code of the table. You can assume the table code is in the range of [01 - 99]. The second field defines the size of the table. You can assume it is an integer > 0.
- Two fields are separated by a space character. You can assume lines are sorted by the table code in the file and the file contains at least one row. Table codes may not be consecutive.

For example, a text file (file1.txt) contains the following:

```
01 2
02 2
03 4
04 4
05 12
06 12
```

This file defines 6 tables. Tables 01 and 02 are two-person tables, and tables 03 and 04 are four-person tables, etc.

Below is the program output for loading data from the input file (Bold red texts are user input):

```
Table file:
file1.txt
Imported 6 table(s).
```

- The above program has NOT terminated yet.
- There is NO space character after the colon (:) and full stop (.).
- Assume the input file and your program are placed in the same folder. You only need to input the filename instead of the full path of the file.

Main menu – accept commands from the user

After loading data from the input file, the system needs to accept commands continuously until "Exit" is received. Including the "Exit" command, you need to implement five commands for the system.

Exit – terminate the program

If the user enters "Exit", the program prints "Bye" and then terminates.

```
Table file:
file1.txt
Imported 6 table(s).
Exit
Bye
```

Book – request a booking

The user can use the "Book" command to request a booking. The user input provides the contact name, phone member, dining date, and the number of seats needed. After receiving the booking, the system should assign it a ticket code. As mentioned, ticket codes for each day are unique, and it counts starting from 1. Please refer to the sample output below for the exact output format.

- The date format in this assignment is DD-MM-YYYY.
- The contact name can contain space characters between English letters.
- You can assume the phone number is 8-digit long.

- Each data field in the user input is separated by a vertical bar (|). The "|" key is usually located below the backspace key in a standard keyboard layout.

Upon completion, the system goes back to the main menu to accept the next command.

In the test case below, 3 bookings are added to the system.

```
Table file:
file1.txt
Imported 6 table(s).
Book|CK Lai|91234567|01-04-2022|10
Added booking. The ticket code for 01-04-2022 is 1.
Book|Tom|61234567|01-04-2022|3
Added booking. The ticket code for 01-04-2022 is 2.
Book|Jolly|51234567|02-04-2022|5
Added booking. The ticket code for 02-04-2022 is 1.
Exit
Bye
```

ListBookings

This command lists all the bookings in the system. Each line shows one booking. For each booking, the contact name, phone number, date with ticket code, number of seats needed, and the booking status are printed.

- If there is no booking in the system, print "No booking.".
- The listing of bookings follows the ordering of user input. That means a newer booking is printed after the older one.
- For a pending booking, print "Pending".
- For a confirmed booking, the codes of the allocated tables are printed. The listing of table codes should be sorted ascendingly. Refer to the next section (AllocateTable) for details and the output format.

Upon completion, the system goes back to the main menu.

```
Table file:
file1.txt
Imported 6 table(s).
ListBookings
No booking.
Book|HF|81234567|01-04-2022|5
Added booking. The ticket code for 01-04-2022 is 1.
Book|Liu|21234567|01-04-2022|2
Added booking. The ticket code for 01-04-2022 is 2.
ListBookings
Booking(s):
HF, 81234567, 01-04-2022 (Ticket 1), 5, Pending.
Liu, 21234567, 01-04-2022 (Ticket 2), 2, Pending.
Exit
Bye
```

AllocateTable

This command allows the manager to allocate tables to a booking. Once tables are allocated, the booking will be changed to confirmed state. In the user input, the dining date and the ticket code are provided to identify the booking, followed by codes of tables to be allocated. After allocation, the system reports allocated tables.

- Table codes are listed in ascending order in the program output, regardless of what order the user input is used.

After allocation and printing, the system goes back to the main menu.

In the test case below, tables 02, 03, 04 are allocated to the booking (Ticket 1) on 01-04-2022.

```
Table file:
file1.txt
Imported 6 table(s).
Book|CK Lai|91234567|01-04-2022|10
Added booking. The ticket code for 01-04-2022 is 1.
Book|Jia|61234567|01-04-2022|3
Added booking. The ticket code for 01-04-2022 is 2.
AllocateTable|01-04-2022|1|04 02 03
Allocated table(s). 01-04-2022 (Ticket 1): 02, 03, 04.
ListBookings
Booking(s):
CK Lai, 91234567, 01-04-2022 (Ticket 1), 10, Assigned table(s): 02, 03, 04.
Jia, 61234567, 01-04-2022 (Ticket 2), 3, Pending.
Exit
Bye
```

You need to handle the following exceptional cases. You can assume NO MORE THAN ONE case would happen simultaneously in our test cases.

- If tables have been allocated to this booking, no further allocation to this booking can be made again. In this case, the system prints “Error: Table(s) already allocated to this booking.“.
- If a table is allocated to another booking on the same day, it cannot be used for this booking. In this case, print “Error: One or more tables allocated to another booking.”.
- If the total seats of suggested tables are not enough for the need of the booking, the system should abort the table allocation and print “Error: Not enough seats for this booking.”.

In the test case below, a certain table is allocated to CK Lai’s booking. Therefore, the system aborts the second attempt of table allocation to the same booking.

```
Table file:
file1.txt
Imported 6 table(s).
Book|CK Lai|91234567|01-04-2022|2
Added booking. The ticket code for 01-04-2022 is 1.
AllocateTable|01-04-2022|1|01
Allocated table(s). 01-04-2022 (Ticket 1): 01.
AllocateTable|01-04-2022|1|02
Error: Table(s) already allocated to this booking.
Exit
```

```
Bye
```

In the test case below, table 01 is allocated to CK Lai's booking (Ticket 1) on 01-04-2022. Therefore, it cannot be allocated to Donald's booking (Ticket 2) on the same day.

```
Table file:
file1.txt
Imported 6 table(s).
Book|CK Lai|91234567|01-04-2022|2
Added booking. The ticket code for 01-04-2022 is 1.
Book|Donald|71234567|01-04-2022|6
Added booking. The ticket code for 01-04-2022 is 2.
AllocateTable|01-04-2022|1|01
Allocated table(s). 01-04-2022 (Ticket 1): 01.
AllocateTable|01-04-2022|2|01 03
Error: One or more tables allocated to another booking.
AllocateTable|01-04-2022|2|02 03
Allocated table(s). 01-04-2022 (Ticket 2): 02, 03.
Exit
Bye
```

In the test case below, Snow White requested a booking for 8 persons to dine with seven little dwarfs. In the first attempt of table allocation, tables 01 and 02 can only provide 4 seats in total, which is not enough for the booking. Therefore, the system aborts the table allocation. In the second attempt, table 05 can provide 12 seats, which is enough for 8 persons. Thus, for this attempt, the system accepts the allocation.

```
Table file:
file1.txt
Imported 6 table(s).
Book|Snow White|12345678|10-04-2022|8
Added booking. The ticket code for 10-04-2022 is 1.
AllocateTable|10-04-2022|1|01 02
Error: Not enough seats for this booking.
AllocateTable|10-04-2022|1|05
Allocated table(s). 10-04-2022 (Ticket 1): 05.
Exit
Bye
```

ListTableAllocation

This command lists the status of table allocation on a particular date. In the user input, the date to be enquired is provided.

- The listing of tables are sorted by table code.
- If the table is allocated to a booking, print the ticket code.
- If the table is available for allocation, print "Available".

```
Table file:
file1.txt
```

```
Imported 6 table(s).
Book|Alex|31234567|01-04-2022|3
Added booking. The ticket code for 01-04-2022 is 1.
AllocateTable|01-04-2022|1|01 02
Allocated table(s). 01-04-2022 (Ticket 1): 01, 02.
ListTableAllocation|01-04-2022
Table(s) on 01-04-2022:
01: Ticket 1
02: Ticket 1
03: Available
04: Available
05: Available
06: Available
Exit
Bye
```

Notes

1. You can assume user inputs and input files are always valid. That means you do not need to handle cases that are not mentioned in the requirement.
2. Input files are used to define table codes and their size only. Do not make any changes to the file from your program.
3. You can use any built-in python functions.

Submission

Submit your program electronically using the Moodle system to **Assignment 4 – Difficult version** under the Assignment section. Late submission will not be accepted.

- Your program must follow the format of the sample input and output strictly.
- The sample input files (file1.txt and myTables.txt) are included in the evaluation environment for test cases evaluation. You do not need to upload them.
- You should only submit source code (*.py)
- We will grade your program with **another set of input files and test cases** (Not limited to the provided sample input files and test cases).

Policy on “Plagiarism” according to the General Office

Plagiarism is a very serious academic offence. Students should understand what constitutes plagiarism, the consequences of committing an offence of plagiarism, and how to avoid it.

As defined in the University's Regulations Governing Conduct at Examinations, plagiarism is "the unacknowledged use, as one's own, of work of another person, whether or not such work has been published.", or put it simply, plagiarism is copying (including paraphrasing) the work of another person (including an idea or argument) without proper acknowledgement.

In case of queries on plagiarism, students are strongly advised to refer to "What is Plagiarism” (https://tl.hku.hk/plagiarism/).

If a student commits plagiarism, with evidence after investigation, no matter whether the student concerned admits or not, a penalty will be imposed:

- First Attempt: if the student commits plagiarism (in an assignment/test of a CS course) for the first time in his/her entire course of study, the student shall be warned in writing and receive zero mark for the whole assignment or the whole test; if the student does not agree, s/he can appeal to the BEng(CompSc) Programme Director within a week.
- Subsequent Attempt: if the student commits plagiarism more than once in higher course of study, the case shall be referred to the Programme Director for consideration. The Programme Director shall investigate the case and consider referring it to the University Disciplinary Committee, which may impose any of the following penalties: a published reprimand, suspension of study for a period of time, fine, or expulsion from the University.

Both the student who copies other's work and the student who offers his/her work for copying shall be penalized.

Teachers should report plagiarism cases to the General Office for records and the issuing of warning letters.