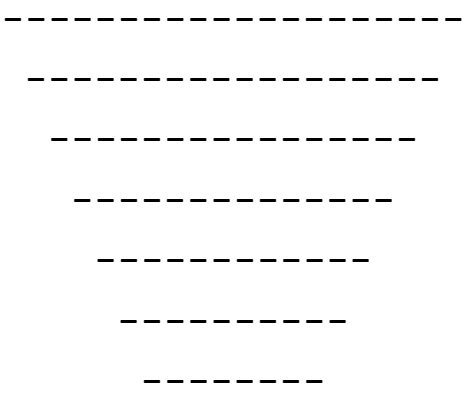
Theatre Ticketing System

Program Description



Your task is to develop a program that handles the ticket sales of a fictional theatre. The layout of the theatre is shown in the above figure. **This layout is fixed**: the first row has 8 seats, the second row has 10 seats, the third row has 12 seats, the fourth row has 14 seats, the fifth row has 16 seats, the sixth row has 18 seats, and the seventh row has 20 seats. The program has two types of users: admin and normal users. Upon starting the program, the users will be asked whether they are admin on not. Based on their choice, the program will allow different operations.

Normal User

If the user is a normal user, in the beginning they will be asked the following question:

```
Are you a new user? (Y/N)
```

If the user enters, Y, the program will ask them to register by providing a new username:

```
Please enter your preferred username: Alice
```

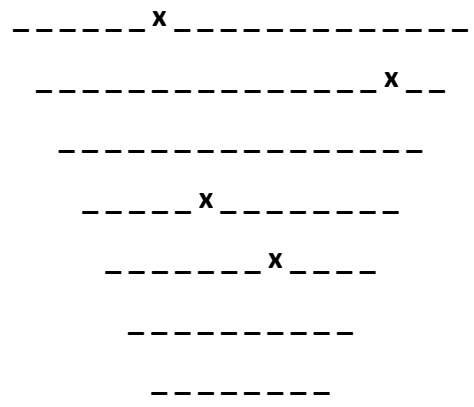
The program will then check whether that new username exists already, if not, it will register the user and show him/her different ticketing options. If the username exists, the program will ask the user to choose a new username and continue this process until a unique username is chosen.

If the user says that they are not a new user, the program will ask him/her to enter their username. It will then check whether the username is registered. If not, it will continue to ask the user to enter the correct username. Otherwise, the program shows them the following menu:

1. Book ticket
2. Cancel ticket
3. Show ticket
4. Quit

Note that, the above menu is also shown to the user when they successfully register.

If a user chooses book ticket, they will see the current booking status for the theatre (the already booked seats are marked with the symbol **x**), and asked to choose a seat:

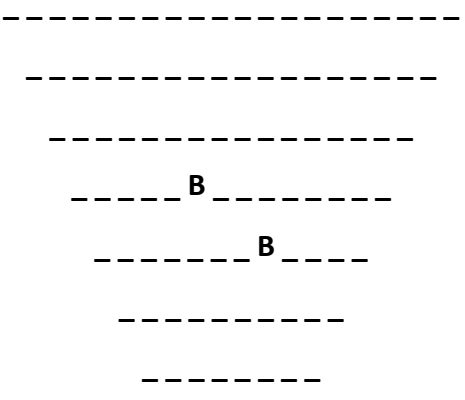



```
Please enter the row: 1
Please enter the seat number: 2
```

Note that the seat number in all the rows starts from 1. Once the user enters their choice, the program checks whether that seat is already booked. If it is booked, it will show a message that the seat is already booked and continue ask them again to choose a seat until a valid seat is chosen. Once a valid seat is chosen, the program shows the price of the seat to user and asks them to confirm their choice. Once confirmed, the booking is recorded. The price is based on the following table:

Row Number	Price (GBP)
1	100
2	80
3	70
4	70
5	60
6	40
7	20

If the user chooses, cancel ticket, they will be shown the theatre layout, along with the seats they have booked:



After that the program asks the user to select a seat that they want to cancel. Once, entered, the program will check whether the seat is booked by the person. If not, it will ask to enter a valid seat number. If a valid seat number is entered, the program will show the ticket information in the following format:

Row: 3

Seat: 8

Price: £70

Do you want to cancel? (Y/N)

If the user says yes, the ticket will be cancelled.

If the user chooses show tickets, the program will show all the tickets, bought by the user, in the following format,

```
Ticket 1

Row: 3

Seat: 8

Price: £70

----------------

Ticket 2

Row: 1

Seat: 2

Price: £100
```

In the case shown above, the user has bought two tickets.

Finally, if the user chooses to quit the program, the program terminates.

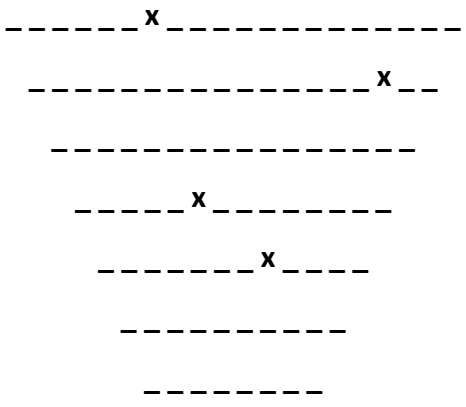
IMPORTANT: Please note that once a user starts the program again, they should be able to see the lates ticketing status and the tickets they have booked. That is, if someone else bought some tickets after a user quits, when that user starts the program again, they will see this updated information when they choose book ticket option. They will also see their previously bought tickets when they chose to cancel or show ticket.

Admin User

If the user says that they are an admin user, the program shows them the following menu:

1. View Ticketing Status
2. Cancel ticket
3. Reset
4. Quit

The view ticketing status option shows the admin a layout of the theatre where all the bought tickets are marked with symbol **x**.



The cancel ticket option allows the admin to cancel any tickets. It first shows the ticket status to the admin. After that the program asks the admin to select a seat that they want to cancel. Once, entered, the program will check whether the seat is booked by any person. If not, it will ask to enter a valid seat number. If a valid seat number is entered, the program will show the ticket information in the following format:

Row: 3

Seat: 8

Price: £70

Do you want to cancel? (Y/N)

If the admin says yes, the ticket will be cancelled.

The reset option is like the cancel ticket option, but in this case all the booked tickets are cancelled. Note that, before cancelling, the program confirms this again from the admin.

Are you sure you want to reset? (Y/N)

If the admin says yes, the ticket will be cancelled.

Finally, if the admin chooses to quit the program, the program terminates.

Marking Criteria

Milestone 1 (10% of module)

For milestone 1, you must submit an outline of all the functions. You do not need to write actual code in the function body. However, the functions must have parameters and comments about their purpose. You should submit the .py file(s) containing the function outline. The .py file(s) should be zipped and uploaded in Moodle. **This should accompany with verbal explanations in the lab session in week 11 and 12. If you cannot be there on weeks 10 and 11, please apply for EC.**

The deadline for this submission is 7th April 2024.

Milestone 2 (30% of module)

For milestone 2, you must complete your code and submit it. Even if your program does not have all the functionalities, you can submit. The submission should have **two files**: one pdf file that will have all your codes and one zip file that will contain all the source codes (*.py files). The final code will be evaluated based on the following criteria:

1. **Functionalities (30 marks):** We will assess how many functionalities, which are discussed above, are there in your program.
2. **Files (20 marks):** Your program should work if a user quits and enters again. This can be done using files. We will assess whether you have used files or not.
3. **Functions (20 marks):** We will assess whether you have organized your codes using functions.
4. **Modules (10 marks):** We will assess whether you have organized your functions in modules.
5. **Exception Handling (10 marks):** Your program must handle exceptions and should not terminate abruptly.
6. **Checking Input Error (10 marks):** The users can provide wrong input at various points in the program. You must check this and show them appropriate messages.

The deadline for this submission is 5th May 2024.