Question 1
This program should be designed and written by a team of students. Here are some suggestions:
• One student should design function main , which will call the other functions in the program. The remainder of the functions will be designed by other members of the team.
• The requirements of the program should be analyzed so each student is given about the same work load.
• The parameters and return types of each function should be decided in advance.
• The program can be implemented as a multi-file program, or all the functions can be cut and pasted into the main file.
Here is the assignment: Write a program that can be used by a small theater to sell tickets for performances. The theater's auditorium has 15 rows of seats, with 30 seats in each row. The program should display a screen that shows which seats are available and which are taken. For example, the following screen shows a chart depicting each seat in the theater. Seats that are taken are represented by an * symbol, and seats that are available are represented by a # symbol:

```
                Seats
          123456789012345678901234567890
Row 1     ***###***###*##########*******###
Row 2     ###*************###********##
Row 3     **###**********#########****###
Row 4     **#######################******
Row 5     ********#####**********########
Row 6     ############**************###
Row 7     ######**************##########
Row 8     ************##****############
Row 9     ########****#####################***
Row 10    #####**************##########
Row 11    #**********##################**
Row 12    ############*********########*
Row 13    ##************########**######
Row 14    ##############################
Row 15    ##############################
```

Here is a list of tasks this program must perform:
• When the program begins, it should ask the user to enter the seat prices for each row. The prices can be stored in a separate array. (Alternatively, the prices may be read from a file.)
• Once the prices are entered, the program should display a seating chart similar to the one shown above. The user may enter the row and seat numbers for tickets being

sold. Every time a ticket or group of tickets is purchased, the program should display the total ticket prices and update the seating chart.
• The program should keep a total of all ticket sales. The user should be given an option of viewing this amount.
• The program should also give the user an option to see a list of how many seats have been sold, how many seats are available in each row, and how many seats are available in the entire auditorium.
Input Validation: When tickets are being sold, do not accept row or seat numbers that do not exist. When someone requests a particular seat, the program should make sure that seat is available before it is sold.

Question 2

ITA1001	ITA1002	ITA1003	ITA1004	ITA1005	ITA1006
ITA2001	ITA2002	ITA2003	ITA2004	ITA2005	ITA2006
ITE1001	ITE1002	ITE1003	ITE1004	ITE1005	ITE1006
ITE2001	ITE2002	ITE2003	ITE2004	ITE2005	ITE2006
SWE1001	SWE1002	SWE1003	SWE1004	SWE1005	SWE1006
SWE2001	SWE2002	SWE2003	SWE2004	SWE2005	SWE2006

The table mentioned above contains the course codes of courses that are to be offered in the next semester. A faculty can give 5 preferences from within the list for course allocation. Get the preferences through command line arguments. If a faculty fails to give his/her preferences, then the Time table committee may fill the preferences by choosing any 5 courses randomly from the list. If a faculty gives less than 5 preferences, the left out ones will be filled randomly. Write a program to implement the same and print the 5 preferences at last.
[Hint: Generate 5 random numbers within the range 0-35 as there are a total of 36 courses in the table. Use the generated random number to find out the row number and column number of the table to choose the course. Example: if the generated number is 17, row number=17/6=2 and column number=17%6=5. Therefore the course ITE1006 should be used for filling.]

Question 3
A small airline has just purchased a computer for its new automated reservations system. You've been asked to develop the new system. You're to write an application to assign seats on each flight of the airline's only plane (capacity: 10 seats). Your application should display the following alternatives: Please type 1 for First Class and Please type 2 for Economy. If the user types 1, your application should assign a seat in the first class section (seats 1–5). If the user types 2, your application should assign a seat in the economy section (seats 6–10). Your application should then display a boarding pass indicating the person's seat number and whether it's in the first-class or economy section of the plane. Use a one-dimensional array of primitive type boolean to represent the seating chart of the plane. Initialize all the elements of the array to false to indicate that all the seats are empty. As each seat is assigned, set the corresponding element of the array to true to indicate that the seat is no longer available.
Your application should never assign a seat that has already been assigned. When the economy section is full, your application should ask the person if it's acceptable to be placed in the first-class section (and vice versa). If yes, make the appropriate seat assignment. If no, display the message "Next flight leaves in 3 hours."