**Problem statement:** Write a program to assign passengers seats in an airplane. Assume a small airplane with seat numbering as follows:

1 A B C D

2 A B C D

3 A B C D

4 A B C D

5 A B C D

6 A B C D

7 A B C D

The program should display the seat pattern, with an ‘X’ marking the seats already assigned. After displaying the seats available, the program prompts the seat desired, the user types in a seat, and then the display of available seats is updated. This continues until all seats are filled or until the user signals that the program should end. If the user types in a seat that is already assigned, the program should say that the seat is occupied and ask for another choice.

**Input Example:**

For example, after seats 1A, 2B, and 4C are taken, the display should look like:

1 X B C D

2 A X C D

3 A B C D

4 A B X D

5 A B C D

6 A B C D

7 A B C D

**Solution**

The whole problem can be implemented with help of 4 major functions:

1. **getData()**
2. **display()**
3. **check()**
4. **update()**

The entire problem is discussed dividing into parts focusing on functional modularity.

**1. Initialize a 2-D array to represent the seat matrix**

A 2-D character array is used to represent the seat matrix where the first column have the row number of each seat & the rest of the columns have four seat A,B,C,D respectively. Thus it’s a 7X5 2-D array to represent the airplane seat matrix which looks like following:

**2. Take user input for seat no**

User is requested to input the desired seat no by giving corresponding no of seat. Like a valid seat no is 1A, 3C, 7D and so on, whereas, an invalid seat request can be 6F, 0B so on.

The input is taken by our function **getData()** which takes user input & returns the string.

**3. Check the seat no request (check() )**

Our major function for this problem is to check the validity of the seat request & update seat matrix status as per request.

* Firstly it checks whether the user input is in the range 1A to 7D or not. If user input is out of range a prompt out **"invalid request"** & continue for further request.
* Check whether user input is **'N'** or not. If it's **'N'** then user wants to end the program. Terminate.
* if seat request is valid but already occupied  
  Then prompt a message stating “It’s already occupied”  
  This checking can be done by founding the 2-D array row & column index for the corresponding seat.  
  Let, **row\_index=r&column\_index=c  
  If(seat\_matrix[row\_index][column\_index]==’X’)**  
  Seat is occupied.
* **ELSE** seat request is a valid one and not occupied still  
  **Update()**

**4. Update the valid entry**

If the request is to update the valid seat we simple change its value to **'X'**. It can be done by finding row & column index and updating the value of **seat\_matrix** at that location to **'X'**.

**5. Special function to check whether all seats are occupied**

The program also need to be terminated when all seats are occupied. Thus every time we keep a checking whether all the entry of **seat\_matrix** is **'X'** or not.

//Airline Seat Reservation Problem//

#include <bits/stdc++.h>

using namespace std;

// to check whether all sits are occupied or not

int allOccupied(char arr[7][5]){

int count=0;

for(int i=0;i<7;i++){

for(int j=1;j<5;j++)

if(arr[i][j]=='X')

count++;

}

if(count==28)

return 1;

return 0;

}

//to display the sits

void display(char arr[7][5]){

for(int i=0;i<7;i++){

for(int j=0;j<5;j++){

cout<<arr[i][j]<<" ";

}

cout<<endl;

}

return;

}

//take user data

string getData(){

string p;

cout<<"enter valid seat no to check(like 1B) or N to end: ";

cin>>p;

return p;

}

//update sit status

void update(char arr[7][5],int row,int col){

cout<<"congrats, your seat is valid. Reserved for you\n";

cout<<"updated seat status..........\n";

arr[row][col]='X';

}

//checking whether user request for

//his sit no can be processed or not

int check(char arr[7][5],string s){

//if user input is not in the range 1A to 7D

if(s[0]>'7' || s[0]<'1' || s[1]>'D' || s[1]<'A'){

cout<<"invalid seat no\n"; //invalid sit no

return 0;

}

int row=-1,col=-1;

//find the row no of the user sit

for(int i=0;i<7;i++){

if(arr[i][0]==s[0])

row=i;

}

//find the column no of user sit

for(int j=0;j<5;j++){

if(arr[row][j]==s[1])

col=j;

}

//check whether sit is already occupied or not.

if(col==-1){

cout<<"Seat is already occupied\n";

return 0;

}

else{

//if it's a valid sit & not occupied,

//process the requested & update the sit as occupied

update(arr,row,col);

}

return 1;

}

void airline(char arr[7][5]){

// user can stop process by pressing 'N'

cout<<"enter N if you are done!\n";

string s;

// continue if not interrepted by user or

//there is valid sit in unoccupied state

while(true){

s=getData(); //get user input

//if user input is to stop the process

if(s[0]=='N')

break; // break

//process the request & check according to

if(check(arr,s))

display(arr);

if(allOccupied(arr)){ //if all sits are occupied

cout<<"sorry, no more seats left!!!!!!!!!!1..."<<endl;

break; //break

}

}

cout<<"Thanks, that's all"<<endl; //end of program

}

int main()

{

//2-D array for storing sit number

char arr[7][5];

for(int i=0;i<7;i++){

//forst column is row number

arr[i][0]=i+1+'0';

for(int j=1;j<5;j++){

//to represent sit number A,B,C,D respectively

arr[i][j]='A'+j-1;

}

}

cout<<"initial seat arrangements........\n";

display(arr);

airline(arr); //airline function

return 0;

}