Note: Before running Hostel\_Allotment\_System.cpp, please run Make\_Rooms\_File.cpp

**1) Hostel\_Allotment\_System.cpp**

#include<iostream>

#include<conio.h>

#include<fstream>

#include<stdio.h>

#include<stdlib.h>

#include<dos.h>

#include<iomanip>

#include <queue>

using namespace std;

class room

{

public:

int AC1, NAC1, AC2, NAC2, AC3, NAC3, AC4, NAC4, AC6, NAC6;

};

room A, B, C, D, E, F, G, H, J, K, L, Q;

class hostel

{

char block;

string roomtype;

int room\_no;

public:

string name;

string reg\_no;

int Rank;

void main\_menu();

void add();

void display();

void rooms();

void edit();

int check(string);

void modify(string);

void delete\_rec(string);

};

queue <hostel> ranking;

void displayTable();

void mergeSort(hostel arr[], int l, int r);

void make\_queue();

void merge(hostel arr[], int l, int m, int r);

void assign\_hostel\_room\_availabilty();

int modify\_hostel\_room\_availability(int flag, char b, string r);

void save\_hostel\_room\_availability();

void hostel::main\_menu()

{

int choice;

while(choice!=5)

{

system("cls");

displayTable();

cout<<"\n\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n\t\t\t\tHOSTEL MANAGEMENT ";

cout<<"\n\t\t\t\t \* MAIN MENU \*";

cout<<"\n\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n\n\n\t\t\t1.Room Allotment";

cout<<"\n\t\t\t2.View Student Details";

cout<<"\n\t\t\t3.Rooms Allotted";

cout<<"\n\t\t\t4.Edit Record";

cout<<"\n\t\t\t5.Exit";

cout<<"\n\n\t\t\tEnter Your Choice: ";

cin>>choice;

switch(choice)

{

case 1:

add();

break;

case 2:

display();

break;

case 3:

rooms();

break;

case 4:

edit();

break;

case 5:

save\_hostel\_room\_availability();

break;

default:

{

cout<<"\n\n\t\t\tWrong choice.....!!!";

cout<<"\n\t\t\tPress any key to continue....!!";

getch();

}

}

}

}

void hostel::add()

{

system("cls");

if(ranking.empty())

{

cout<<"\n\tNo more students left...!!!!"<<endl;

cout<<"\n\n\tPress any key to continue.....!!";

getch();

return;

}

displayTable();

int flag = 1;

char b;

string r;

cout<<"\n\t"<<"THE DETAILS OF THE CURRENT STUDENT ARE: "<<endl;

cout<<"\t"<<"NAME: "<<(ranking.front()).name<<endl;

cout<<"\t"<<"REG NO: "<<(ranking.front()).reg\_no<<endl;

cout<<"\t"<<"RANK: "<<(ranking.front()).Rank<<endl<<endl;

name = (ranking.front()).name;

reg\_no = (ranking.front()).reg\_no;

Rank = (ranking.front()).Rank;

cout<<"Enter the block: ";

cin>>b;

cout<<"Enter the room type(1AC/1NAC/2AC/2NAC/3AC/3NAC/4AC/4NAC/6AC/6NAC): ";

cin>>r;

flag = modify\_hostel\_room\_availability(1, b, r);

if(flag==1)

{

block = b;

roomtype=r;

cout<<"Room no.: ";

cin>>room\_no;

ofstream fout("Record",ios::app);

fout.write((char\*)this,sizeof(hostel));

fout.close();

cout<<"\n Room is booked...!!!";

ranking.pop();

}

cout<<"\n Press any key to continue.....!!";

getch();

}

void hostel::display()

{

system("cls");

ifstream fin("Record");

int flag=0;

string r;

cout<<"\nEnter the registration number of the student: ";

cin>>r;

while(fin.read((char\*)this,sizeof(hostel)))

{

if(reg\_no==r)

{

system("cls");

cout<<"\n Student Details:";

cout<<"\n\n Room no: "<<room\_no;

cout<<"\n Name: "<<name;

cout<<"\n Reg No: "<<reg\_no;

cout<<"\n Block: "<<block;

cout<<"\n Room type: "<<roomtype;

flag=1;

break;

}

}

if(flag==0)

cout<<"\n Sorry registration number not found !!";

cout<<"\n\n Press any key to continue....!!";

getch();

fin.close();

}

void hostel::rooms()

{

system("cls");

ifstream fin("Record");

cout<<"\n\t\t\t\t\t List Of Rooms Allotted"<<endl;

cout<<"\n\t\t\t\t -------------------------------------------"<<endl;

cout<<" "<<endl;

cout<<setw(20)<<"Room No."<<setw(5)<<"|"<<setw(15)<<"Name"<<setw(5)<<"|"<<setw(15)<<"Reg No"<<setw(5)<<"|"<<setw(10)<<"Block"<<setw(5)<<"|"<<setw(15)<<"Room Type"<<setw(5)<<"|"<<endl;

cout<<setw(20)<<" ------------|-------------------|-------------------|--------------|-------------------|"<<endl;

while(fin.read((char\*)this,sizeof(hostel)))

{

cout<<setw(20)<<room\_no<<setw(5)<<"|"<<setw(15)<<name<<setw(5)<<"|"<<setw(15)<<reg\_no<<setw(5)<<"|"<<setw(10)<<block<<setw(5)<<"|"<<setw(15)<<roomtype<<setw(5)<<"|"<<endl;

}

cout<<"\n\n\n\t\t\tPress any key to continue.....!!";

getch();

fin.close();

}

void hostel::edit()

{

system("cls");

int choice;

string r;

cout<<"\n EDIT MENU:";

cout<<"\n\n 1.Modify Student Record";

cout<<"\n 2.Delete Student Record";

cout<<"\n Enter your choice: ";

cin>>choice;

system("cls");

cout<<"\nEnter registration number of the student: " ;

cin>>r;

switch(choice)

{

case 1:

modify(r);

break;

case 2:

delete\_rec(r);

break;

default:

cout<<"\n Wrong Choice.....!!";

}

cout<<"\n Press any key to continue....!!!";

getch();

}

int hostel::check(string r)

{

int flag=0;

ifstream fin("Record");

while(fin.read((char\*)this,sizeof(hostel)))

{

if(reg\_no==r)

{

flag=1;

break;

}

}

fin.close();

return(flag);

}

void hostel::modify(string r)

{

int flag=0;

ifstream fin("Record");

ofstream fout("temp");

while(fin.read((char\*)this,sizeof(hostel)))

{

if(reg\_no==r){

modify\_hostel\_room\_availability(0, block, roomtype);

cout<<endl;

displayTable();

cout<<"\nEnter New Details:";

cout<<"\n\nEnter the block: ";

cin>>block;

cout<<"Enter the room type(1AC/1NAC/2AC/2NAC/3AC/3NAC/4AC/4NAC/6AC/6NAC): ";

cin>>roomtype;

cout<<"Room no.: ";

cin>>room\_no;

fout.write((char\*)this,sizeof(hostel));

modify\_hostel\_room\_availability(1, block, roomtype);

flag=1;

}

else{

fout.write((char\*)this,sizeof(hostel));

}

}

fin.close();

fout.close();

if(flag==0)

cout<<"\n Sorry registration number not found !!";

else

{

remove("Record");

rename("temp","Record");

}

}

void hostel::delete\_rec(string r)

{

int flag=0;

char ch;

ifstream fin("Record");

ofstream fout("temp");

while(fin.read((char\*)this,sizeof(hostel)))

{

if(reg\_no==r)

{

cout<<"\n Name: "<<name;

cout<<"\n Registration number: "<<reg\_no;

cout<<"\n Room number: "<<room\_no;

cout<<"\n Block: "<<block;

cout<<"\n Room type: "<<roomtype;

cout<<"\n\n Do you want to delete this record(y/n): ";

cin>>ch;

if((ch=='n')||(ch=='N'))

fout.write((char\*)this,sizeof(hostel));

else

modify\_hostel\_room\_availability(0, block, roomtype);

flag=1;

}

else

fout.write((char\*)this,sizeof(hostel));

}

fin.close();

fout.close();

if(flag==0)

cout<<"\n Sorry registration number not found !!";

else

{

remove("Record");

rename("temp","Record");

}

}

void displayTable()

{

cout<<"\t"<<"Block 1-Bed 2-Bed 3-Bed 4-Bed 6-Bed"<<endl;

cout<<"\t"<<"\tAC\tNAC\t AC\tNAC\t AC\tNAC\t AC\tNAC\t AC\tNAC"<<endl;

cout<<"\t"<<"A\t"<<A.AC1<<"\t"<<A.NAC1<<"\t "<<A.AC2<<"\t"<<A.NAC2<<"\t "<<A.AC3<<"\t"<<A.NAC3<<"\t "<<A.AC4<<"\t"<<A.NAC4<<"\t "<<A.AC6<<"\t"<<A.NAC6<<" "<<endl;

cout<<"\t"<<"B\t"<<B.AC1<<"\t"<<B.NAC1<<"\t "<<B.AC2<<"\t"<<B.NAC2<<"\t "<<B.AC3<<"\t"<<B.NAC3<<"\t "<<B.AC4<<"\t"<<B.NAC4<<"\t "<<B.AC6<<"\t"<<B.NAC6<<" "<<endl;

cout<<"\t"<<"C\t"<<C.AC1<<"\t"<<C.NAC1<<"\t "<<C.AC2<<"\t"<<C.NAC2<<"\t "<<C.AC3<<"\t"<<C.NAC3<<"\t "<<C.AC4<<"\t"<<C.NAC4<<"\t "<<C.AC6<<"\t"<<C.NAC6<<" "<<endl;

cout<<"\t"<<"D\t"<<D.AC1<<"\t"<<D.NAC1<<"\t "<<D.AC2<<"\t"<<D.NAC2<<"\t "<<D.AC3<<"\t"<<D.NAC3<<"\t "<<D.AC4<<"\t"<<D.NAC4<<"\t "<<D.AC6<<"\t"<<D.NAC6<<" "<<endl;

cout<<"\t"<<"E\t"<<E.AC1<<"\t"<<E.NAC1<<"\t "<<E.AC2<<"\t"<<E.NAC2<<"\t "<<E.AC3<<"\t"<<E.NAC3<<"\t "<<E.AC4<<"\t"<<E.NAC4<<"\t "<<E.AC6<<"\t"<<E.NAC6<<" "<<endl;

cout<<"\t"<<"F\t"<<F.AC1<<"\t"<<F.NAC1<<"\t "<<F.AC2<<"\t"<<F.NAC2<<"\t "<<F.AC3<<"\t"<<F.NAC3<<"\t "<<F.AC4<<"\t"<<F.NAC4<<"\t "<<F.AC6<<"\t"<<F.NAC6<<" "<<endl;

cout<<"\t"<<"G\t"<<G.AC1<<"\t"<<G.NAC1<<"\t "<<G.AC2<<"\t"<<G.NAC2<<"\t "<<G.AC3<<"\t"<<G.NAC3<<"\t "<<G.AC4<<"\t"<<G.NAC4<<"\t "<<G.AC6<<"\t"<<G.NAC6<<" "<<endl;

cout<<"\t"<<"H\t"<<H.AC1<<"\t"<<H.NAC1<<"\t "<<H.AC2<<"\t"<<H.NAC2<<"\t "<<H.AC3<<"\t"<<H.NAC3<<"\t "<<H.AC4<<"\t"<<H.NAC4<<"\t "<<J.AC6<<"\t"<<H.NAC6<<" "<<endl;

cout<<"\t"<<"J\t"<<J.AC1<<"\t"<<J.NAC1<<"\t "<<J.AC2<<"\t"<<J.NAC2<<"\t "<<J.AC3<<"\t"<<J.NAC3<<"\t "<<J.AC4<<"\t"<<J.NAC4<<"\t "<<J.AC6<<"\t"<<J.NAC6<<" "<<endl;

cout<<"\t"<<"K\t"<<K.AC1<<"\t"<<K.NAC1<<"\t "<<K.AC2<<"\t"<<K.NAC2<<"\t "<<J.AC3<<"\t"<<K.NAC3<<"\t "<<K.AC4<<"\t"<<K.NAC4<<"\t "<<K.AC6<<"\t"<<K.NAC6<<" "<<endl;

cout<<"\t"<<"L\t"<<L.AC1<<"\t"<<L.NAC1<<"\t "<<L.AC2<<"\t"<<L.NAC2<<"\t "<<K.AC3<<"\t"<<L.NAC3<<"\t "<<L.AC4<<"\t"<<L.NAC4<<"\t "<<L.AC6<<"\t"<<L.NAC6<<" "<<endl;

cout<<"\t"<<"Q\t"<<Q.AC1<<"\t"<<Q.NAC1<<"\t "<<Q.AC2<<"\t"<<Q.NAC2<<"\t "<<Q.AC3<<"\t"<<Q.NAC3<<"\t "<<Q.AC4<<"\t"<<Q.NAC4<<"\t "<<Q.AC6<<"\t"<<Q.NAC6<<" "<<endl;

}

void merge(hostel arr[], int l, int m, int r)

{

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

hostel L[n1], R[n2];

for (i = 0; i < n1; i++)

L[i] = arr[l + i];

for (j = 0; j < n2; j++)

R[j] = arr[m + 1+ j];

i = 0;

j = 0;

k = l;

while (i < n1 && j < n2)

{

if (L[i].Rank <= R[j].Rank)

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

void mergeSort(hostel arr[], int l, int r)

{

if (l < r)

{

int m = l+(r-l)/2;

mergeSort(arr, l, m);

mergeSort(arr, m+1, r);

merge(arr, l, m, r);

}

}

void make\_queue()

{

int i, n;

system("cls");

cout<<"Enter the number of students(enter 0 if you just want to view or edit the records): ";

cin>>n;

hostel data[n];

cout<<"\n\n";

if(n!=0)

cout<<"Enter the name, registration number and rank of students: "<<endl<<endl;

for(i=0; i<n; i++)

{

cout<<"Name: ";

cin>>data[i].name;

cout<<"Registration number: ";

cin>>data[i].reg\_no;

cout<<"Rank: ";

cin>>data[i].Rank;

cout<<endl;

}

mergeSort(data, 0, n-1);

for(i=0; i<n; i++)

{

ranking.push(data[i]);

}

}

void assign\_hostel\_room\_availabilty()

{

ifstream fin("Rooms");

fin.read((char\*)&A,sizeof(A));

fin.read((char\*)&B,sizeof(B));

fin.read((char\*)&C,sizeof(C));

fin.read((char\*)&D,sizeof(D));

fin.read((char\*)&E,sizeof(E));

fin.read((char\*)&F,sizeof(F));

fin.read((char\*)&G,sizeof(G));

fin.read((char\*)&H,sizeof(H));

fin.read((char\*)&J,sizeof(J));

fin.read((char\*)&K,sizeof(K));

fin.read((char\*)&L,sizeof(L));

fin.read((char\*)&Q,sizeof(Q));

fin.close();

}

int modify\_hostel\_room\_availability(int flag, char b, string r)

{

int return\_flag=1;

if((b=='A')&&(r=="1AC")&&(A.AC1>0||flag==0))

A.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='A')&&(r=="1NAC")&&(A.NAC1>0||flag==0))

A.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='A')&&(r=="2AC")&&(A.AC2>0||flag==0))

A.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='A')&&(r=="2NAC")&&(A.NAC2>0||flag==0))

A.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='A')&&(r=="3AC")&&(A.AC3>0||flag==0))

A.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='A')&&(r=="3NAC")&&(A.NAC3>0||flag==0))

A.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='A')&&(r=="4AC")&&(A.AC4>0||flag==0))

A.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='A')&&(r=="4NAC")&&(A.NAC4>0||flag==0))

A.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='A')&&(r=="6AC")&&(A.AC6>0||flag==0))

A.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='A')&&(r=="6NAC")&&(A.NAC6>0||flag==0))

A.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="1AC")&&(B.AC1>0||flag==0))

B.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="1NAC")&&(B.NAC1>0||flag==0))

B.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="2AC")&&(B.AC2>0||flag==0))

B.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="2NAC")&&(B.NAC2>0||flag==0))

B.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="3AC")&&(B.AC3>0||flag==0))

B.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="3NAC")&&(B.NAC3>0||flag==0))

B.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="4AC")&&(B.AC4>0||flag==0))

B.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="4NAC")&&(B.NAC4>0||flag==0))

B.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="6AC")&&(B.AC6>0||flag==0))

B.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='B')&&(r=="6NAC")&&(B.NAC6>0||flag==0))

B.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="1AC")&&(C.AC1>0||flag==0))

C.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="1NAC")&&(C.NAC1>0||flag==0))

C.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="2AC")&&(C.AC2>0||flag==0))

C.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="2NAC")&&(C.NAC2>0||flag==0))

C.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="3AC")&&(C.AC3>0||flag==0))

C.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="3NAC")&&(C.NAC3>0||flag==0))

C.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="4AC")&&(C.AC4>0||flag==0))

C.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="4NAC")&&(C.NAC4>0||flag==0))

C.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="6AC")&&(C.AC6>0||flag==0))

C.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='C')&&(r=="6NAC")&&(C.NAC6>0||flag==0))

C.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="1AC")&&(D.AC1>0||flag==0))

D.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="1NAC")&&(D.NAC1>0||flag==0))

D.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="2AC")&&(D.AC2>0||flag==0))

D.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="2NAC")&&(D.NAC2>0||flag==0))

D.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="3AC")&&(D.AC3>0||flag==0))

D.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="3NAC")&&(D.NAC3>0||flag==0))

D.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="4AC")&&(D.AC4>0||flag==0))

D.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="4NAC")&&(D.NAC4>0||flag==0))

D.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="6AC")&&(D.AC6>0||flag==0))

D.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='D')&&(r=="6NAC")&&(D.NAC6>0||flag==0))

D.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="1AC")&&(E.AC1>0||flag==0))

E.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="1NAC")&&(E.NAC1>0||flag==0))

E.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="2AC")&&(E.AC2>0||flag==0))

E.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="2NAC")&&(E.NAC2>0||flag==0))

E.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="3AC")&&(E.AC3>0||flag==0))

E.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="3NAC")&&(E.NAC3>0||flag==0))

E.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="4AC")&&(E.AC4>0||flag==0))

E.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="4NAC")&&(E.NAC4>0||flag==0))

E.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="6AC")&&(E.AC6>0||flag==0))

E.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='E')&&(r=="6NAC")&&(E.NAC6>0||flag==0))

E.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="1AC")&&(F.AC1>0||flag==0))

F.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="1NAC")&&(F.NAC1>0||flag==0))

F.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="2AC")&&(F.AC2>0||flag==0))

F.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="2NAC")&&(F.NAC2>0||flag==0))

F.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="3AC")&&(F.AC3>0||flag==0))

F.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="3NAC")&&(F.NAC3>0||flag==0))

F.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="4AC")&&(F.AC4>0||flag==0))

F.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="4NAC")&&(F.NAC4>0||flag==0))

F.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="6AC")&&(F.AC6>0||flag==0))

F.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='F')&&(r=="6NAC")&&(F.NAC6>0||flag==0))

F.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="1AC")&&(G.AC1>0||flag==0))

G.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="1NAC")&&(G.NAC1>0||flag==0))

G.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="2AC")&&(G.AC2>0||flag==0))

G.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="2NAC")&&(G.NAC2>0||flag==0))

G.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="3AC")&&(G.AC3>0||flag==0))

G.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="3NAC")&&(G.NAC3>0||flag==0))

G.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="4AC")&&(G.AC4>0||flag==0))

G.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="4NAC")&&(G.NAC4>0||flag==0))

G.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="6AC")&&(G.AC6>0||flag==0))

G.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='G')&&(r=="6NAC")&&(G.NAC6>0||flag==0))

G.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="1AC")&&(H.AC1>0||flag==0))

H.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="1NAC")&&(H.NAC1>0||flag==0))

H.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="2AC")&&(H.AC2>0||flag==0))

H.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="2NAC")&&(H.NAC2>0||flag==0))

H.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="3AC")&&(H.AC3>0||flag==0))

H.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="3NAC")&&(H.NAC3>0||flag==0))

H.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="4AC")&&(H.AC4>0||flag==0))

H.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="4NAC")&&(H.NAC4>0||flag==0))

H.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="6AC")&&(H.AC6>0||flag==0))

H.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='H')&&(r=="6NAC")&&(H.NAC6>0||flag==0))

H.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="1AC")&&(J.AC1>0||flag==0))

J.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="1NAC")&&(J.NAC1>0||flag==0))

J.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="2AC")&&(J.AC2>0||flag==0))

J.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="2NAC")&&(J.NAC2>0||flag==0))

J.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="3AC")&&(J.AC3>0||flag==0))

J.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="3NAC")&&(J.NAC3>0||flag==0))

J.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="4AC")&&(J.AC4>0||flag==0))

J.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="4NAC")&&(J.NAC4>0||flag==0))

J.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="6AC")&&(J.AC6>0||flag==0))

J.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='J')&&(r=="6NAC")&&(J.NAC6>0||flag==0))

J.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="1AC")&&(K.AC1>0||flag==0))

K.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="1NAC")&&(K.NAC1>0||flag==0))

K.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="2AC")&&(K.AC2>0||flag==0))

K.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="2NAC")&&(K.NAC2>0||flag==0))

K.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="3AC")&&(K.AC3>0||flag==0))

K.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="3NAC")&&(K.NAC3>0||flag==0))

K.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="4AC")&&(K.AC4>0||flag==0))

K.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="4NAC")&&(K.NAC4>0||flag==0))

K.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="6AC")&&(K.AC6>0||flag==0))

K.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='K')&&(r=="6NAC")&&(K.NAC6>0||flag==0))

K.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="1AC")&&(L.AC1>0||flag==0))

L.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="1NAC")&&(L.NAC1>0||flag==0))

L.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="2AC")&&(L.AC2>0||flag==0))

L.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="2NAC")&&(L.NAC2>0||flag==0))

L.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="3AC")&&(L.AC3>0||flag==0))

L.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="3NAC")&&(L.NAC3>0||flag==0))

L.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="4AC")&&(L.AC4>0||flag==0))

L.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="4NAC")&&(L.NAC4>0||flag==0))

L.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="6AC")&&(L.AC6>0||flag==0))

L.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='L')&&(r=="6NAC")&&(L.NAC6>0||flag==0))

L.NAC6 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="1AC")&&(Q.AC1>0||flag==0))

Q.AC1 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="1NAC")&&(Q.NAC1>0||flag==0))

Q.NAC1 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="2AC")&&(Q.AC2>0||flag==0))

Q.AC2 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="2NAC")&&(Q.NAC2>0||flag==0))

Q.NAC2 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="3AC")&&(Q.AC3>0||flag==0))

Q.AC3 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="3NAC")&&(Q.NAC3>0||flag==0))

Q.NAC3 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="4AC")&&(Q.AC4>0||flag==0))

Q.AC4 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="4NAC")&&(Q.NAC4>0||flag==0))

Q.NAC4 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="6AC")&&(Q.AC6>0||flag==0))

Q.AC6 -= (flag==1)?(1):(-1) ;

else if((b=='Q')&&(r=="6NAC")&&(Q.NAC6>0||flag==0))

Q.NAC6 -= (flag==1)?(1):(-1) ;

else

{

cout<<"Room not available!!";

return\_flag = 0;

}

return return\_flag;

}

void save\_hostel\_room\_availability()

{

ofstream fout("temp");

fout.write((char\*)&A,sizeof(A));

fout.write((char\*)&B,sizeof(B));

fout.write((char\*)&C,sizeof(C));

fout.write((char\*)&D,sizeof(D));

fout.write((char\*)&E,sizeof(E));

fout.write((char\*)&F,sizeof(F));

fout.write((char\*)&G,sizeof(G));

fout.write((char\*)&H,sizeof(H));

fout.write((char\*)&J,sizeof(J));

fout.write((char\*)&K,sizeof(K));

fout.write((char\*)&L,sizeof(L));

fout.write((char\*)&Q,sizeof(Q));

fout.close();

remove("Rooms");

rename("temp","Rooms");

}

int main()

{

hostel h;

assign\_hostel\_room\_availabilty();

system("cls");

cout<<"\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n\t\t\t\* HOSTEL ALLOTMENT SYSTEM \*";

cout<<"\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"\n\n\t\tDeveloped By:";

cout<<"\t Tanish Malekar, Parth Kulkarni and Adya Agrawal";

cout<<"\n\n\n\n\n\n\n\t\t\t\t\tPress any key to continue....!!";

getch();

make\_queue();

cout<<"\n\n\n\n\t\t\t\t\tPress any key to continue....!!";

getch();

h.main\_menu();

return 0;

}

**2) Make\_Rooms\_File.cpp**

#include <iostream>

#include<fstream>

using namespace std;

class room

{

public:

int AC1, NAC1, AC2, NAC2, AC3, NAC3, AC4, NAC4, AC6, NAC6;

};

room A, B, C, D, E, F, G, H, J, K, L, Q;

int main()

{

A.AC1=1; A.NAC1=1; A.AC2=0; A.NAC2=4; A.AC3=0; A.NAC3=0; A.AC4=6; A.NAC4=4; A.AC6=0; A.NAC6=3;

B.AC1=1; B.NAC1=0; B.AC2=1; B.NAC2=0; B.AC3=6; B.NAC3=7; B.AC4=0; B.NAC4=1; B.AC6=0; B.NAC6=18;

C.AC1=0; C.NAC1=0; C.AC2=0; C.NAC2=8; C.AC3=0; C.NAC3=0; C.AC4=0; C.NAC4=6; C.AC6=0; C.NAC6=0;

D.AC1=64; D.NAC1=94; D.AC2=0; D.NAC2=0; D.AC3=0; D.NAC3=0; D.AC4=0; D.NAC4=0; D.AC6=0; D.NAC6=0;

E.AC1=0; E.NAC1=0; E.AC2=0; E.NAC2=4; E.AC3=0; E.NAC3=0; E.AC4=0; E.NAC4=0; E.AC6=0; E.NAC6=0;

F.AC1=0; F.NAC1=0; F.AC2=6; F.NAC2=3; F.AC3=0; F.NAC3=0; F.AC4=0; F.NAC4=6; F.AC6=0; F.NAC6=0;

G.AC1=8; G.NAC1=233; G.AC2=0; G.NAC2=0; G.AC3=0; G.NAC3=0; G.AC4=7; G.NAC4=2; G.AC6=0; G.NAC6=0;

H.AC1=9; H.NAC1=1; H.AC2=0; H.NAC2=0; H.AC3=0; H.NAC3=7; H.AC4=0; H.NAC4=0; H.AC6=0; H.NAC6=2;

J.AC1=2; J.NAC1=8; J.AC2=0; J.NAC2=0; J.AC3=0; J.NAC3=785; J.AC4=0; J.NAC4=0; J.AC6=0; J.NAC6=6;

K.AC1=0; K.NAC1=0; K.AC2=0; K.NAC2=0; K.AC3=0; K.NAC3=0; K.AC4=6; K.NAC4=0; K.AC6=5; K.NAC6=0;

L.AC1=1; L.NAC1=0; L.AC2=4; L.NAC2=2; L.AC3=6; L.NAC3=0; L.AC4=3; L.NAC4=0; L.AC6=6; L.NAC6=0;

Q.AC1=0; Q.NAC1=0; Q.AC2=7; Q.NAC2=0; Q.AC3=4; Q.NAC3=0; Q.AC4=8; Q.NAC4=0; Q.AC6=9; Q.NAC6=0;

ofstream fout("Rooms");

fout.write((char\*)&A,sizeof(A));

fout.write((char\*)&B,sizeof(B));

fout.write((char\*)&C,sizeof(C));

fout.write((char\*)&D,sizeof(D));

fout.write((char\*)&E,sizeof(E));

fout.write((char\*)&F,sizeof(F));

fout.write((char\*)&G,sizeof(G));

fout.write((char\*)&H,sizeof(H));

fout.write((char\*)&J,sizeof(J));

fout.write((char\*)&K,sizeof(K));

fout.write((char\*)&L,sizeof(L));

fout.write((char\*)&Q,sizeof(Q));

fout.close();

}