QUẢN LÝ KHÁCH SẠN

#### Tạo struct **customer** chứa thông tin khách hàng{tên,ID,năm sinh,tuổi,quê quán }

struct customer

{

char name[20];

char ID[15];

int year;

int age;

char home[20];

}

#### Tạo struct in**room** chứa thông tin phòng

struct inforoom

{

**customer khach;**

int bed: (phòng 1 giường đơn/phòng 2 giường đôi)

int price;

bool status: (OutofOrder/Occupied) - true/false

bool vip: true/false

int bill=0

}

#### class **motel**

{

inroom **room[**N**];** dùng array

int **money** (doanh thu)**;**

Create();

Rent(customer x,int to); (thuê)

*Pay(int from); ??*

oneday(); //

Show();

Change(); (đổi phòng)

}

#### **motel** khởi tạo phòng()

{

for(int i=1;i<số phòng;i++)

loại phòng=??;

if(loại phòng=??)

{

số giường=??;

giá tiền =phụ thuộc vào loại và số giường??;

}

else{}

tình trạng=false;

}

#### **motel** thuê phòng(customer x,int to)

{

scanf(i) // nhập số phòng

if(x.tuổi>=18)

if(phòng[i].tinhtrang==false) tức khi phòng có trạng thái trống

room phòng[i].customer=x;

phòng[i].tinhtrang=true

else in(đã có người);

else

nothing

in(ch đủ tuổi);

}

#### **customer** cusinfo(customer x)

{

gets(x.tên);

x.tên=checkname(x.tên) //chuẩn hóa tên

gets(x.ID);

x.ID=checkid(x.ID)//kiểm tra và nhập ID;

scanf(năm sinh);

x.namsinh=kiemtra(x.namsinh)//kiểm tra để năm sinh trong khoảng nhất định

gets(quê);

x.tuổi=2021-x.namsinh

return x;

}

motel oneday()

{

cộng tiền vào bill của phòng;(mỗi ngày);

//vậy là hàm này mình duyệt từ 1-50, phòng nào có người đang ở thì thêm vào hàm newBill(i)

}

newBill(i){

{

//cần tạo 4 biến lưu 4 giá tiền ở struct

switch(i) xem i thuộc khoảng nào rồi:

phong[i].price += 1 trong 4 biến

}

#### 

#### motel trả phòng(int from)

{

scanf(i) // nhập số phòng

**kiểm tra tình trạng phòng r tiến hành**

doanhthu=doanhthu+phòng[i].bill; (nếu bill=0 thì cộng vào price)

// này là chưa tính tiền số ngày ở của phòng, nên thêm 1 biến hóa đơn từng phòng khi trả r cộng vào doanh thu, chứ dt này là dt tổng của ks(đã thêm 1day() để pass từng ngày)

phòng[i].customer=NULL;

phòng[i].tình trạng=false;

}

display(int i)

{

kiểm tra nếu phòng có người(kiểm tra bằng tình trạng phòng) thì in số phòng và thông tin

in lần lượt thông tin

}

motel show()

{

chạy từ 1-50

display(i);

}

motel chuyển phòng()

{

nhập phòng chuyển(from) và phòng chuyển tới(to)

kiểm tra trạng thái 2 phòng{from có khách và to trống}

lưu khách hàng vào customer x

pay(from);

rent(to);

các case còn lại in thông báo tương ứng

}

### Các hàm cần hoàn thiện:

* Chuẩn hóa tên: truyền vào địa chỉ chuỗi sau đó format ngay trên chuỗi
* Kiểm tra ID: yêu cầu nhập vào ID và kiểm tra, hàm trả về int{kiểm tra trùng có thể dùng find}
* Kiểm tra năm sinh: yêu cầu nhập namsinh và ktra, trả về int{năm nên nằm khoảng 1950-2007 vì cần 14t mới có ID}
* đọc file:
* ghi file:
* switch case của main
* xóa khách hàng trong hàm trả phòng{tạo 1 customer x rỗng và thay vào}hoặc đơn giản là thay đổi status phòng
* chuyển phòng{swap customer 2 phòng đồng thời thanh toán phòng trước đó}
* ~~hiển thị{in thông tin khách hàng}~~

#### Lưu ý:

##### Quy ước loại phòng dựa trên số phòng:

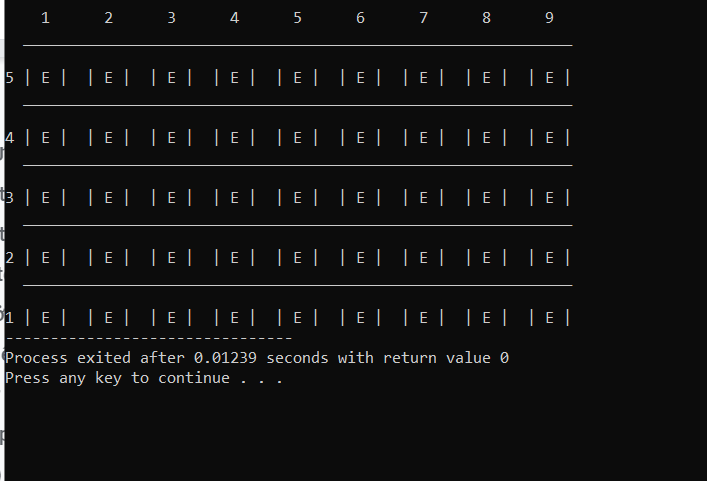
01-10: bình thường, giường đơn - 200k/1 ngày đêm

11-20: bình thường giường đôi - 400k/1 ngày đêm

21-30: vip, giường đơn - 500k/1 ngày đêm

31-40: vip, giường đôi - 1tr/ 1 ngày đêm

Hiển thị trực quan

****