Dsa-Lab-Task : 12

Roll No: 24P-0706 **Dept: BS-CS**

Name: Aazan Noor Khuwaja **Section : 3D**

Qns:

Code:

```
#include<iostream>

using namespace std;

class order{

public:

string item_name,cust_name;

int price,quantity,priority;




void get_data();

void display_data();
```

```
//order(string n,int p,int q,string c_n,int
prio):item_name(n),cust_name(c_n),price(p),quantity(q),priority(prio){};

};



void order::get_data()

{

cout<<"Enter The item Name:"<<endl;

cin>>item_name;

cout<<"Enter The Customer Name:"<<endl;

cin>>cust_name;

cout<<"Enter The Price:"<<endl;

cin>>price;

cout<<"Enter Quantity:"<<endl;

cin>>quantity;

cout<<"Enter The priority:"<<endl;

cin>>priority;

}

void order::display_data()

{

cout<<"The item Name:"<<item_name<<endl;

cout<<"The Customer Name:"<<cust_name<<endl;

cout<<"The Price:"<<price<<endl;

cout<<"The Quantity:"<<quantity<<endl;
```

```
cout<<"The priority:"<<priority<<endl;
}


class heap{
private:
void create_heapify_up(int i)
{
order temp=o1[i];
while(i>1 && temp.priority>o1[i/2].priority)
{
o1[i]=o1[i/2];
i=i/2;
}
o1[i]=temp;
}
void create_heapify_down(int i)
{
int left_child=left(i);
int right_child=right(i);
int largest=i;
// Check if left child exist and has higher priority or not then will set largest according that
if (left_child<=orders && o1[left_child].priority>o1[largest].priority)
{
```

```
largest =left_child;
}
// Check if right child exist and has higher priority
if (right_child<=orders && o1[right_child].priority>o1[largest].priority)
{
largest=right_child;
}
// If largest is not the current , swap and continue heapifying down
if (largest!=i)
{
swap(o1[i], o1[largest]);
create_heapify_down(largest);
}
}

public:

order o1[30];
int orders;
heap():orders(0){}
```

```
int left(int i){
return 2*i;
}
int right(int i){
return 2*i+1;
}
int parent(int i){
return i/2;
}


void insert_single_order()
{
orders++;
if(orders<=30){
o1[orders].get_data();
create_heapify_up(orders);
}
else{
cout<<"The Heap is Full first free its space!"<<endl;
}
}
```

```
void serve_order()
{
if(orders==0)
{
cout<<"currently no orders are there to serve!\n";
return;
}

cout<<"serving highest priority order:\n";
o1[1].display_data(); //root order
o1[1]=o1[orders]; //move last order to root
orders--; //reduce heap size
create_heapify_down(1); //restore max-heap

cout<<"Highest Priority order Served!"<<endl;
}

void display_orders()
{
```

```
if(orders==0)
{
cout<<"Currently No order available !"<<endl;
return;
}
for(int i=1;i<=orders;i++)
{
cout<<"Order "<<i<<": \n";
o1[i].display_data();
cout<<endl;
}
}
};


int main()
{
heap h;
int choice;
do
{
cout<< "\n1.Put New Order\n2.Serve highest priority order\n3. Display all orders currently avalaible\n4.Exit\n";
cin >>choice;
switch(choice)
```

```
{
case 1: h.insert_single_order(); break;
case 2: h.serve_order(); break;
case 3: h.display_orders(); break;
case 4: cout << "Exiting...\n"; break;
default: cout << "Invalid choice!\n";
}
} while(choice != 4);
}
```

```
TERMINAL  PORTS  SPELL CHECKER 11  GITHUB  PROBLEMS 11  OPEN EDITORS  DEBUG CONSOLE  OUTPUT

[aazannoorkhuwaja@archlinux ~/Cs_Data/3rd_Semester/DSA_lab/lab 12]$ g++ -g restaurant_mng_system_heap.cpp -o ./out
[aazannoorkhuwaja@archlinux ~/Cs_Data/3rd_Semester/DSA_lab/lab 12]$ ./out

 1.Put New Order
 2.Serve highest priority order
 3. Display all orders currently avalaible
 4.Exit
 1
 Enter The item Name:
 Stick
 Enter The Customer Name:
 Aazan
 Enter The Price:
 50
 Enter Quantity:
 3
 Enter The priority:
 1

 1.Put New Order
 2.Serve highest priority order
 3. Display all orders currently avalaible
 4.Exit
 2
 serving highest priority order:
 The item Name:Stick
 The Customer Name:Aazan
 The Price:50
 The Quantity:3
 The priority:1

 1.Put New Order
 2.Serve highest priority order
 3. Display all orders currently avalaible
 4.Exit
 3
 Currently No order available !

 1.Put New Order
 2.Serve highest priority order
 3. Display all orders currently avalaible
 4.Exit
 1
 Enter The item Name:
```

```
TERMINAL  PORTS  SPELL CHECKER 11  GITHUB  PROBLEMS 11  OPEN EDITORS  DEBUG CONSOLE  OUTPUT

[aazannoorkhuwaja@archlinux ~/Cs_Data/3rd_Semester/DSA_lab/lab 12]$ ./out
Enter The item Name:
chips
Enter The Customer Name:
ghilman
Enter The Price:
50
Enter Quantity:
3
Enter The priority:
1

1.Put New Order
2.Serve highest priority order
3. Display all orders currently avalaible
4.Exit
1
Enter The item Name:
stick
Enter The Customer Name:
ank
Enter The Price:
30
Enter Quantity:
1
Enter The priority:
9

1.Put New Order
2.Serve highest priority order
3. Display all orders currently avalaible
4.Exit
1
Enter The item Name:
shulttle
Enter The Customer Name:
musawar
Enter The Price:
25
Enter Quantity:
4
Enter The priority:
2
```

```
[aazannoorkhuwaja@archlinux ~/Cs_Data/3rd_Semester/DSA_lab/lab 12]$ ./out
Enter The priority:
2

1.Put New Order
2.Serve highest priority order
3. Display all orders currently avalaible
4.Exit
3
Order 1:
The item Name:stick
The Customer Name:ank
The Price:30
The Quantity:1
The priority:9

Order 2:
The item Name:chips
The Customer Name:ghilman
The Price:50
The Quantity:3
The priority:1

Order 3:
The item Name:shulttle
The Customer Name:musawar
The Price:25
The Quantity:4
The priority:2


1.Put New Order
2.Serve highest priority order
3. Display all orders currently avalaible
4.Exit
1
Enter The item Name:
Peanus
Enter The Customer Name:
javed
Enter The Price:
80
Enter Quantity:
6
```

This is from last :