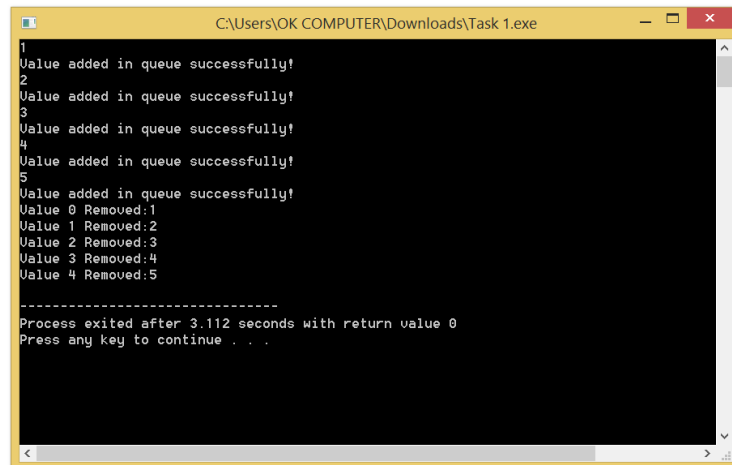


DS LAB 07 QUEUES

TASK 1

```
1  #include<iostream>
2  using namespace std;
3  class Queue{
4  public:
5      int front;
6      int rear;
7      int size;
8      int number;
9      int array[100];
10 Queue(){
11     size=100;
12     front=-1;
13     rear=-1;
14     number=0;
15 }
16 bool QueueCapacity(){
17 if(number<size){
18     return false;
19 }
20 else{
21     return true;
22 }
23 }
24 bool isEmpty(){
25 if(number>0){
26     return false;}
27 else
28     return true;
29 }
30 void AddMember(int x){
31 if(!QueueCapacity()){
32     rear=(rear+1)%size;
33     array[rear]=x;
34     number++;
35 }
36 }
37 int RemoveMember(){
38 if(!isEmpty()){
39     front=(front+1)%size;
40     int x=array[front];
41     number--;
42     return x;
43 }
44 }
45 };
46
47 int main(){
48     Queue q1;
49     int x;
50     for(int i=0;i<5;i++){
51         cin>>x;
52         q1.AddMember(x);
53         cout<<"Value added in queue successfully!"<<endl;
54     }
55     for(int i=0;i<5;i++){
56         cout<<"Value "<<i<<" Removed:"<<q1.RemoveMember()<<endl;
57     }
58 }
```



```
1 Value added in queue successfully!
2 Value added in queue successfully!
3 Value added in queue successfully!
4 Value added in queue successfully!
5 Value added in queue successfully!
Value 0 Removed:1
Value 1 Removed:2
Value 2 Removed:3
Value 3 Removed:4
Value 4 Removed:5
-----
Process exited after 3.112 seconds with return value 0
Press any key to continue . . .
```

TASK 2

```

1  #include<iostream>
2  using namespace std;
3  class node{
4  public:
5  int data;
6  node *next;
7  node *front;
8  node *rear;
9  int size;
10 int number;
11 node(){
12     size=100;
13     front=NULL;
14     rear=NULL;
15     number=0;
16     next=NULL;
17     data=0;
18 }
19 bool QueueCapacity(){
20 if(number<size){
21     return false;
22 }
23 else{
24     return true;
25 }
26 }
27 void AddMember(int value){
28     node *temp=new node();
29     if(rear==NULL){
30         node *temp=new node();
31         if(rear==NULL){
32             rear=temp;
33             rear->next=NULL;
34             rear->data=value;
35             front=rear;
36         }
37         else{
38             rear->next=temp;
39             temp->data=value;
40             temp->next=NULL;
41             rear=temp;
42         }
43         number++;
44     }
45 }
46 int RemoveMember(){
47 if(front==NULL){
48     cout<<"Queue Underflow!";
49 }
50 int x=front->data;
51 front=front->next;
52 number--;
53 return x;
54 }
55 };
56
57 int x=front->data;
58 front=front->next;
59 number--;
60 return x;
61 }
62 };
63
64 int main(){
65     node n1;
66     int x;
67     for(int i=0;i<5;i++){
68         cin>>x;
69         n1.AddMember(x);
70         cout<<"Value added in linked list successfully!"<<endl;
71     }
72     for(int i=0;i<5;i++){
73         cout<<"Value "<<i<<" Removed from linked list:"<<n1.RemoveMember()<<endl;
74     }
75 }

```

```

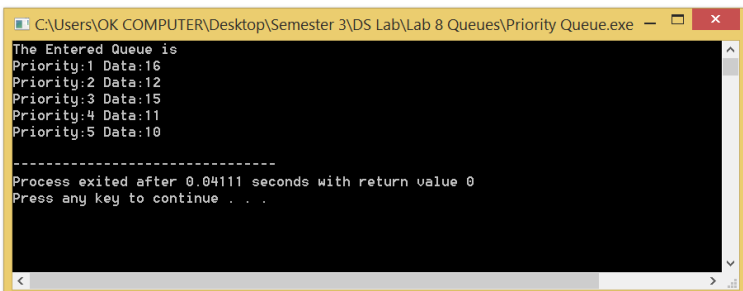
C:\Users\OK COMPUTER\Downloads\Task 2.exe
10
Value added in linked list successfully!
9
Value added in linked list successfully!
8
Value added in linked list successfully!
7
Value added in linked list successfully!
6
Value added in linked list successfully!
Value 0 Removed from linked list:10
Value 1 Removed from linked list:9
Value 2 Removed from linked list:8
Value 3 Removed from linked list:7
Value 4 Removed from linked list:6

```

TASK 3

Priority Queue.cpp Task 1.cpp Task 2.cpp Stack with array.cpp

```
1 #include <iostream>
2 using namespace std;
3 class node{
4 public:
5     int priority;
6     int data;
7     node *next;
8     node(){
9         priority=0;
10        data=0;
11        next=NULL;
12    }
13 };
14 class queue{
15     node *front;
16 public:
17     queue(){
18         front= NULL;
19     }
20     void insert(int i, int p) {
21         node *temp,*q;
22         temp=new node;
23         temp->data = i;
24         temp->priority=p;
25         if (front == NULL || p < front->priority) {
26             temp->next= front;
27             front = temp;
28         } else {
29             q = front;
30             while (q->next != NULL && q->next->priority <= p)
31                 q = q->next;
32             temp->next = q->next;
33             q->next = temp;
34         }
35     }
36     void del() {
37         node *temp;
38         if(front == NULL)
39             cout<<"Queue Underflow\n";
40         else {
41             temp = front;
42             cout<<"Deleted item is: "<<temp->data<<endl;
43             front = front->next;
44             delete temp;
45         }
46     }
47     void show(){
48         node *p;
49         p= front;
50         if (front == NULL)
51             cout<<"Queue is empty\n";
52         else {
53             cout<<"The Entered Queue is"<<endl;
54             while(p != NULL) {
55                 cout<<"Priority:"<<p->priority<<" Data:"<<p->data<<endl;
56                 p = p->next;
57             }
58         }
59     }
60 };
61 int main(){
62     queue q;
63     q.insert(16,1);
64     q.insert(15,3);
65     q.insert(12,2);
66     q.insert(10,5);
67     q.insert(11,4);
68     q.show();
69 }
```



```
C:\Users\OK COMPUTER\Desktop\Semester 3\DS Lab\Lab 8 Queues\Priority Queue.exe
The Entered Queue is
Priority:1 Data:16
Priority:2 Data:12
Priority:3 Data:15
Priority:4 Data:11
Priority:5 Data:10

Process exited after 0.04111 seconds with return value 0
Press any key to continue . . .
```