



**Jawahar Education Societys Annasaheb Chudaman Patil College of  
Engineering, Kharghar, Navi Mumbai**

**NAME: PRIYUSH BHIMRAO KHOBRADE**

**PRN NO: 211112018**

**SUBJECT: DATA STRUCTURE LAB**

PAGE NO.:

DATE.: / / 20

Practical No - 05

Aim :- Implement priority Queue ADT using Array.

Theory :-

Priority Queue

"A Priority Queue is abstract data type that behave similarly to the normal queue except that each element has some priority." i.e. Element with highest priority would come first in priority queue.

However, if elements with same priority occur, they are served according to their order in the queue.

• Types of priority Queue

• Ascending order priority Queue :- "In ascending order priority Queue, a lower priority number is given as higher priority in a priority."

EX. Ascending order like 1, 2, 3, 4, 5 ∴ smallest number i.e 1 is given as the highest priority in priority queue.

• Descending order priority Queue :- "In Descending order priority Queue, a higher priority number is given as higher priority in a priority."

EX. Descending order like 5, 4, 3, 2, 1 ∴ 5 is given highest priority in priority queue.

1

Teachers Signature \_\_\_\_\_

### Algorithm Priority Queue using Array

#### • Enqueue()

1. IF (FRONT == 0) && (REAR == N-1)
2. Print " overflow condition"
3. ELSE
4. IF (FRONT == -1)
5. FRONT = REAR = 0,
6. QUEUE [REAR] = data
7. Priority [REAR] = priority
8. ELSE IF (REAR == N-1)
9. FOR i = FRONT; i < REAR; i++)
10. FOR (i = FRONT; i < REAR; i++)
11. Q[i - FRONT] = Q[i]
12. PR[i - FRONT] = PR[i]
13. REAR = REAR - FRONT
14. FRONT = 0
15. FOR (i = 0; i < REAR; i++)
16. IF (P > PR[i])
17. Q[i+1] = Q[i] & PR[i+1] = PR[i]
18. ELSE
19. Q[i+1] = data & PR[i+1] = P
20. REAR++

PAGE NO.:

DATE: / / 20

### • Dequeue()

1. IF (Front == -1)
2. PRINT "Queue Under Flow Condition"
3. ELSE
4. PRINT "Q [F], P [F]"
5. IF (Front == Rear)
6. Front = Rear = -1
7. ELSE
8. FRONT++

### print()

1. FOR (i = Front ; i <= Rear ; i++)
2. PRINT (Q[i], P[i])

Conclusion :- We can identify the properties on doing task of priority queue. It data structure helps you arrange your data according to your priority and make your task easy.

Teachers Signature \_\_\_\_\_

3

## AIM: Implement Priority Queue ADT using array

### Input:

```
1 #include<stdio.h>
2 #define N 20
3 int Q[N],Pr[N];
4 int r = -1, f = -1;
5 void enqueue(int data,int p)//Enqueue function to insert data and its priority in queue
6 {
7     int i;
8     if((f==0)&&(r==N-1)) //Check if Queue is full
9         printf("Queue is full");
10    else
11    {
12        if(f==-1)//if Queue is empty
13        {
14            f = r = 0;
15            Q[r] = data;
16            Pr[r] = p;
17        }
18        else if(r == N-1)//if there is some elements in Queue
19        {
20            for(i=f;i<=r;i++) { Q[i-f] = Q[i]; Pr[i-f] = Pr[i]; r = r-f; f = 0; for(i = r;i>f;i--)
21                {
22                    if(p>Pr[i])
23                    {
24                        Q[i+1] = Q[i];
25                        Pr[i+1] = Pr[i];
26                    }
27                    else
28                        break;
29                    Q[i+1] = data;
30                    Pr[i+1] = p;
31                    r++;
32                }
33            }
34        }
35        else
36        {
37            for(i = r;i>=f;i--)
38            {
39                if(p>Pr[i])
40                {
41                    Q[i+1] = Q[i];
42                    Pr[i+1] = Pr[i];
43                }
44                else
45                    break;
46            }
47            Q[i+1] = data;
48            Pr[i+1] = p;
49            r++;
50        }
51    }
52 }
53
54 }
55 void print() //print the data of Queue
56 {
57     int i;
58     for(i=f;i<=r;i++)
59     {
60         printf("\nElement = %d\tPriority = %d",Q[i],Pr[i]);
61     }
62 }
63 int dequeue() //remove the data from front
64 {
65     if(f == -1)
66     {
67         printf("Queue is Empty");
68     }
69     else
70     {
71         printf("deleted Element = %d\t Its Priority = %d",Q[f],Pr[f]);
72         if(f==r)
73             f = r = -1;
```

### AIM: Implement Priority Queue ADT using array

```
74         else
75             f++;
76     }
77 }
78 int main()
79 {
80     int opt,n,i,data,p;
81     printf("Enter Your Choice-");
82     do{
83         printf("\n\n1 for Insert the Data in Queue\n2 for show the Data in Queue\n3 for Delete the data from the Queue\n0
for Exit");
84         scanf("%d",&opt);
85         switch(opt){
86             case 1:
87                 printf("\nEnter the number of data");
88                 scanf("%d",&n);
89                 printf("\nEnter your data and Priority of data");
90                 i=0;
91                 while(i<n){
92                     scanf("%d %d",&data,&p);
93                     enqueue(data,p);
94                     i++;
95                 }
96                 break;
97             case 2:
98                 print();
99                 break;
100            case 3:
101                dequeue();
102                break;
103            case 0:
104                break;
105            default:
106                printf("\nIncorrect Choice");
107            }
108        }while(opt!=0);
109    return 0;
110 }
111 }
```

## AIM: Implement Priority Queue ADT using array

### Output:-

"C:\Users\Rupesh\Documents\DS 2ND\Implement Priority Queue ADT using array\_P5.exe"

Enter Your Choice:-

```
1 for Insert the Data in Queue
2 for show the Data in Queue
3 for Delete the data from the Queue
0 for Exit
1
```

Enter the number of data 5

Enter your data and Priority of data

```
54 65
48 69
47 96
33 22
35 95
```

```
1 for Insert the Data in Queue
2 for show the Data in Queue
3 for Delete the data from the Queue
0 for Exit
3
```

deleted Element = 47      Its Priority = 96

```
1 for Insert the Data in Queue
2 for show the Data in Queue
3 for Delete the data from the Queue
0 for Exit
2
```

```
Element = 35      Priority = 95
Element = 48      Priority = 69
Element = 54      Priority = 65
Element = 33      Priority = 22
```

```
1 for Insert the Data in Queue
2 for show the Data in Queue
3 for Delete the data from the Queue
0 for Exit
```

**Conclusion:** - We can identify the properties on doing task of priority queue. It data structure help your arrange your data according to your priority and make your task easy.