



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

COURSE CODE: DJS22ITL302

DATE: 5/10/2023

COURSE NAME: Data Structure Laboratory

CLASS: I1-Batch1

NAME: Anish Sharma

Experiment No. 1

CO/LO: CO1

Aim: Implements Priority Queue using array

Theory: Priority Queue implementation using an array is one of the basic methods to implement a queue. In this element, it is inserted and deleted based on its priority. The elements in the priority queue have some priority. The priority of the element is used to determine the order in which the elements will be processed.

Program:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define MAX 10
```

```
void create_queue();
```

```
void insert_element(int);
```

```
void delete_element(int);
```

```
void check_priority(int);
```

```
void display_priorityqueue();
```

```
int pqueue[MAX];
```



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

int front, rear;

void main()

{

int n, choice;

printf("\nEnter 1 to insert element by priority ");

printf("\nEnter 2 to delete element by priority ");

printf("\nEnter 3 to display priority queue ");

printf("\nEnter 4 to exit");

create_queue();

while (1)

{

printf("\nEnter your choice : ");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("\nEnter element to insert : ");

scanf("%d",&n);

insert_element(n);

break;



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

case 2:

```
printf("\nEnter element to delete : ");
```

```
scanf("%d",&n);
```

```
delete_element(n);
```

```
break;
```

case 3:

```
display_priorityqueue();
```

```
break;
```

case 4:

```
exit(0);
```

default:

```
printf("\n Please enter valid choice");
```

```
}
```

```
}
```

```
}
```

```
void create_queue()
```

```
{
```

```
    front = rear = -1;
```

```
}
```

```
void insert_element(int data)
```

```
{
```



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

```
if (rear >= MAX - 1)

{

    printf("\nQUEUE OVERFLOW");

    return;

}

if ((front == -1) && (rear == -1))

{

    front++;

    rear++;

    pqueue[rear] = data;

    return;

}

else

    check_priority(data);

    rear++;

}

void check_priority(int data)

{

    int i,j;

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

    {
```



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

```
if (data >= pqueue[i])
{
    for (j = rear + 1; j > i; j--)
    {
        pqueue[j] = pqueue[j - 1];
    }
    pqueue[i] = data;
    return;
}

pqueue[i] = data;
}

void delete_element(int data)
{
    int i;

    if ((front== -1) && (rear== -1))
    {
        printf("\nEmpty Queue");
        return;
    }

    for (i = 0; i <= rear; i++)
```



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

```
{  
    if (data == pqueue[i])  
    {  
        for (; i < rear; i++)  
        {  
            pqueue[i] = pqueue[i + 1];  
        }  
        pqueue[i] = -99;  
        rear--;  
        if (rear == -1)  
            front = -1;  
        return;  
    }  
}  
  
printf("\n%d element not found in queue", data);  
}  
  
void display_priorityqueue()  
{  
    if ((front == -1) && (rear == -1))  
    {  
        printf("\nEmpty Queue ");  
    }  
}
```



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

```
    return;
}
for (; front <= rear; front++)
{
    printf(" %d ", pqueue[front]);
}
front = 0;
}
```



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology

Output screenshots:

```
Enter 1 to insert element by priority
Enter 2 to delete element by priority
Enter 3 to display priority queue
Enter 4 to exit
Enter your choice : 1

Enter element to insert : 21

Enter your choice : 1

Enter element to insert : 34

Enter your choice : 1

Enter element to insert : 87

Enter your choice : 2

Enter element to delete : 34

Enter your choice : 1

Enter element to insert : 89

Enter your choice : 3
89 87 21
```

Conclusion:

a priority queue is a versatile data structure that extends the functionality of a standard queue by assigning a priority to each element. Elements with higher priority are dequeued and processed before elements with lower priority.

REFERENCES:

Tutorialpoints



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Information Technology