

Date \_\_\_\_\_

# Priority Queue

```

#include <stdio.h>
#include <stdlib.h>
#define N 5
int queue[5][N]
int front[5] = {0, 0, 0, 0, 0}
int rear[5] = {-1, -1, -1, -1, -1}
int item, pr;
void main()
{
    int ch
    clrscr()
    while(1)
    {
        printf("Priority Queue\n");
        printf(" 1: PQ insert\n");
        printf(" 2: PQ delete\n");
        printf(" 3: PQ display\n");
        printf(" 4: Exit\n");
        printf(" Enter your choice\n");
        scanf("%d", &ch);
        switch(ch)
    }
}

```

case 1:

```

printf("Enter priority number\n")

```

```

scanf("%d", &pr);
if(pr > 0 && pr < 4)
    PQinsert(pr-1);
else
    printf(" only 3 priority exists\n")
}

```

break;

case 2:

pq\_delete();  
break;

case 3:

display();  
break;

case 4:

exit(0);

3) pq\_insert(int px)

```
if [rear(pq)] == n - 1
    printf("Queue overflow\n");
else
    {
        printf("Enter the item\n");
        scanf("%d", &item);
        rear(pq)++;
        queue(pq)[rear(pq)] = item;
    }
return;
```

## Delete()

```

int i;
for(i=0; i<3; i++)
{
    if rear[i] == front[i]-1
        printf("Queue Empty\n");
    else
        printf("Deleted item is %d
of Queue %d\n",
               queue[i], front,
               (i+1));
    front[i]++;
    queue[i];
}
    
```

## display()

```

int i, j;
for(i=0; i<3; i++)
{
    if rear[i] == front[i]-1
        printf("Queue
empty %d\n",
               i);
    else
        { }
}
    
```

```
printf ("Queue %d: ", i+1);  
for (j = front(i); j < rear(i); j++)  
    printf ("%d", queue[i][j]);
```

{

}

solution:

# Input Restricted and output Restricted Queue

```

#include < stdio.h>
#include < stdlib.h>
#define max 5
Void insert_right();
Void insert_left();
Void delete_right();
Void delete_left();
Void display();
Void input();
Void output();
int q[max], left=-1, right=-1;
int main()
{

```

```

    int option;
    printf("1: Input Restricted\n");
    printf("2: Output Restricted\n");
    printf("3: exit\n");
    printf("Enter your choice\n");
    scanf("%d", &option)
}
```

```
switch(option)
```

```
case 1:
```

```
    input();
    break;
```

```
case 2:
```

```
    output();
    break;
```

Case 3:

exit(0)

{

}

void input()

{

int option;

do

{

printf("1: Insert Right\n");

printf("2: Delete Right\n");

printf("3: Delete Left\n");

printf("4: Display\n");

printf("Enter your option\n")

switch(option)

{

case 1:

insert\_right();  
break;

case 2:

delete\_right();  
break;

case 3:

delete\_left();  
break;

case 4:

display();  
break;

Date \_\_\_\_\_

case 5;

exit(0);

{ while (option != 5);

}

void output()

{

int option;

do

{

printf ("1: Delete right\n");

printf ("2: Insert right\n");

printf ("3: Insert left\n");

printf ("4: Display\n");

printf ("Enter your option\n");

scanf ("%d", &amp;option);

switch (option)

{

case 1:

delete\_left();  
break;

case 2:

insert\_right();  
break;

case 3:

insert\_left();  
break;

Case 4:

```
    display();
    break;
```

Case 5:

```
    exit(0);
```

```
{ while (option != 5);
```

```
void insert_right()
```

```
{ int num;
```

```
printf("Enter the number [n]:");
scanf("%d", &num);
```

```
if (left == 0 & right == max - 1) {
    left = right + 1
}
```

```
{
```

```
printf("Queue overflow [n]:");
exit(0);
```

```
}
```

```
if (left == -1 & right == -1)
```

```
left = 0;
```

```
right = 0;
```

```
{
```

Page No.

Date \_\_\_\_\_

else

{

if (right == max - 1)  
    right = 0

else

right--;

}

q[right] = num;

}

Void insert\_left()

{

int num;

printf("Enter the number[a]:");  
scanf("%d", &num);if (left == 0 && right == max - 1) ||  
(left == right)

{

printf("Queue overflowed");  
exit(0);

}

if (left == -1 &amp;&amp; right == -1)

left = 0;

right = 0;

{

Date \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

else

{

if (left == 0)

left = num - 1;

else

left --;

}

q[left] = num;

}

void delete\_llt()

{

if (left == -1)

printf("Queue underflow\n");

exit(0);

printf("Element Deleted is %d\n", q[left]);

if (left == right)

left = -1;

right = -1;

{

else

{

if (left == num - 1)

left = 0;

else

left += 1;

{

Date \_\_\_\_\_

```
void delete_right()
```

```
{ if (left == -1)
```

```
    printf("Queue underflow\n")
```

~~```
else
```~~ printf("Element deleted : %d\n", 0)

```
{ if (left == right)
```

```
    left = -1;
```

```
, right = -1;
```

```
else
```

```
{
```

```
if (right == 0)
```

```
    right = max - 1
```

```
else
```

```
    right = right - 1
```

```
}
```

```
void display()
```

```
int front = left;
```

```
int rear = right;
```

```
{ if (left == -1)
```

```
    printf("Queue Empty\n");
```

```
exit(0);
```

```
{
```

if (front <= rear)

{  
while (front <= rear)

{  
printf ("%d\n", q[front]);  
front++;

{

else  
{

while (front <= max - 1)  
{

printf ("%d\n", q[front]);  
front++;

{

front = 0;  
while (front <= rear)

{

printf ("%d\n", q[front]);  
front++;

{

{

Output

- 1: input Restricted
- 2: output Restricted
- 3: Exit

Enter your choice  
1

- 1: InsertRight
- 2: delete right
- 3: delete left
- 4: display
- 5: exit

Enter

Enter your option  
1

Enter the number

5

Enter yo

- 1: Insert Right
- 2: Delete Right
- 3: Delete left
- 4: display
- 4: Exit

Enter Your option

1

Enter the number  
E