

## LAB-4-Queue Implementation

NAME : CHIRAG A

USN : IBM19CS039

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#define limit 5

int front = -1, rear = -1, queue[limit], flag = 1;

void insert();
int del();
void display();

void main()
{
    int choice, x;
    clrscr();
    while (1)
    {
        printf("\n 1: Insert \n 2: Display \n 3: Delete \n 4: Exit \n");
        printf("\n Enter your choice:");
        scanf("%d", &choice);
        switch (choice) {
            case 1: insert();
                    break;
            case 2: display();
                    break;
            case 3: x = del();
                    if (flag == 0)
                        printf("\n Element cannot be deleted");
```

else

printf("\n Deleted element is %d", x);

break;

case 4: exit(0);

break;

default: printf("\n Invalid choice !!!");

}

}

}

void insert()

{

int item;

if (rear == limit - 1)

{

printf("\n Queue Overflow\n");

return;

}

else

{

printf("\n Enter element to be inserted:");

scanf("%d", &item);

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

front = rear = 0;

else rear++;

queue[rear] = item;

return;

}

}

int del()

{

int item;

```
if (front == -1 && rear == -1)
{
    printf("\n Queue is empty\n");
    flag = 0;
    return -1;
}
else
{
    item = queue[front];
    if (front == rear)
        front = rear = -1;
    else front++;
    return item;
}
}
```

```
void display()
{
    int i;
    if (front == -1 && rear == -1)
    {
        printf("\n Queue is Empty\n");
        return;
    }
    else
    {
        printf("\n");
        for (i = front; i <= rear; i++)
            printf("%d", queue[i]);
    }
}
```