

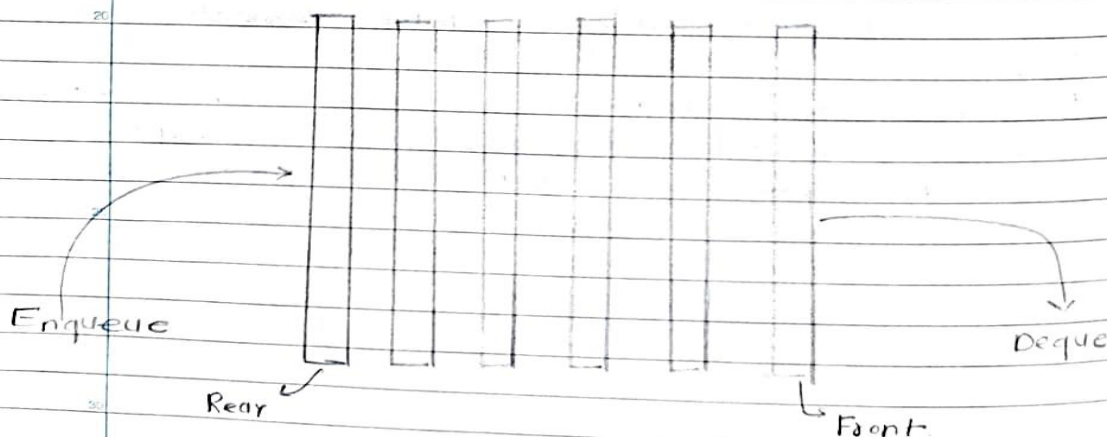
# DSA Experiment No.4

## Experiment No. 04

AIM - Queues are frequently used data structures in the programming, and a typical job is a creation of job by an operating system. If the operating system doesn't use priorities, then the jobs are processed in order they entered the system. Write C++ program for simulating job queue. Write functions to add job and delete job from queue.

### Theory →

A Queue is a linear data structure which follows a particular order in which operations are performed. The order is first in first out (FIFO). A good example for queue is a consumer based example where first come first served for resources. The difference between <sup>is</sup> stacks and queues is in removing. In stack we remove the item most recently added; in a queue we remove the least recently added item.



## Applications of Queue data structure →

Queue is used when things don't have to be processed immediately, but have to be processed in First In First Out order like breadth first search.

This property of queue makes it useful in the following scenarios.

- ① When a resource is shared between multiple users  
ex - include cpu scheduling, Disk scheduling.
- ② When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes, Examples → include IO Buffers, pipes, files IO, etc.
- ③ In operating systems :
  - a) Semaphores
  - b) FCFS (first come first served) scheduling
  - c) Spooling in printers
  - d) Buffer for devices like keyboard.
- ④ In Networks :
  - a) Queues in routers/switches
  - b) Mail Queues.
- ⑤ Variations.: (Deque, priority queue, Doubly ended priority queue).

## Program code:

```
#include <iostream>
#include <conio.h>
#define MAX 10
using namespace std;

struct queue
{
    int data[MAX];
    int front,rear;
};

class Queue
{
    struct queue q;
public:
    Queue(){q.front=q.rear=-1;}
    int isempty();
    int isfull();
    void enqueue(int);
    int delqueue();
    void display();
};

int Queue::isempty()
{
    return(q.front==q.rear)?1:0;
}

int Queue::isfull()
{
    return(q.rear==MAX-1)?1:0;
```

```

}
void Queue::enqueue(int x)
{
    q.data[++q.rear]=x;
}
int Queue::delqueue()
{
    return q.data[++q.front];
}
void Queue::display()
{
    for(int i=q.front+1;i<=q.rear;i++)
        cout<<q.data[i]<<" ";
}
int main()
{
    Queue obj;
    int ch,x;
    cout<<"SCOB86_Rudraksh Karpe\n\n";
    do{
        cout<<"\n To Insert data -> Press 1 \n ";
        cout<<"\n To Delete data -> Press 2 \n";
        cout<<"\n To Display data -> Press 3 \n";
        cout<<"\n To Exit -> Press 4 \n";
        cout<<"\n Enter your choice : ";

        cin>>ch;
        switch(ch)
        {
            case 1:
                if (!obj.isfull())
                {
                    cout<<"\n Enter data : ";

```

```

        cin>>x;
        obj.enqueue(x);
    }
    else
        cout<< "Queue is overflow...\n";
        break;
    case 2:
    if(!obj.isempty())
    {
        cout<<"\n Deleted Element is = "<<obj.delqueue();
    }
    else
    {
        cout<<"\n Queue is underflow...\n";
    }
    cout<<"\nRemaining Elements are : ";
    obj.display();
        break;
    case 3:
    if (!obj.isempty())
    {
        cout<<"\n Queue contains : ";
        obj.display();
    }
    cout<<"\n";
    }
    else
        cout<<"\n Queue is empty...\n";
        break;
    case 4: cout<<"\n Exit...";
    }
}while(ch!=4);

return 0;

```

}

## Output of the program:

### Insertion of elements:

```
PS R:\GHRCEM\DSA Lab\Assignment 4> cd "r:\GHRCEM\DSA Lab\Assignment 4\" ; if ($?) { g++ 4.cpp -o 4 } ; if ($?) { .\4 }
SCOB86_Rudraksh Karpe

Enter data : 2
To Insert data -> Press 1
To Delete data -> Press 2
To Display data -> Press 3
To Exit -> Press 4
Enter your choice : 1
Enter data : 4
To Insert data -> Press 1
To Delete data -> Press 2
To Display data -> Press 3
To Exit -> Press 4
Enter your choice : 3
Queue contains : 2 4
To Insert data -> Press 1
To Delete data -> Press 2
To Display data -> Press 3
To Exit -> Press 4
Enter your choice : █
```

### Deletion of elements:

```
Deleted Element is = 2
Remaining Elements are : 4
To Insert data -> Press 1
To Delete data -> Press 2
To Display data -> Press 3
To Exit -> Press 4
Enter your choice : █
```