

Assignment - 11

Page No.:

Date:

ROLL NO. 21123

Dir. SEI

Batch - F1

QUP:

POS:

Title - Implementing Queue.

Problem Statement - Queue are frequently used in computer Engg' programs & typical example is creation of job queue by an operating system. If the operating system does not use priorities, then jobs are processed in the orders they enter the system. Write C++ program for simulating job queue. function to add job and delete job from queue.

objective: Implement concept of queue in C++.

outcome: student will be able to write & execute C++ program to create a job queue and Queue operations.

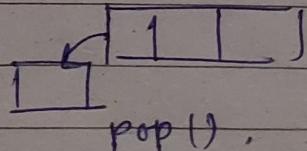
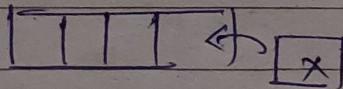
S/I/W & H/I/W operations:

Windows 10, mingw compiler, Eclipse IDE, Intel i5-9300H
4 core, 8 logical processor | 8 GB RAM, 512 GB SSD, NVIDIA
Geforce GTX 1050 Ti GDDR5.

theory: Queue is a data structure that follows FIFO convention. The elements that enters first, leaves queue has 2 main operations.

push back (x)

pop front ()



They are also called enqueue & dequeue operations.

Algorithm ,

- ① Create a class stack which takes the data member as front & rear & array [max]
- ② Create a function for Enqueue &


```
if (rear==max-1) { Overflow condition }
else if (rear== -1 & front== -1) { rear=0; front=0;
array [rear] = x;
}
else { rear++; array [rear] = x; }
```
- ③ Create a function for Dequeue


```
if (front & rear both are -1) { then queue is empty }
else if (rear == front) { rear = front = -1 }
else { decrement rear increment front }.
```
- ④ For displaying the queue ,
 1. Set for loop from $j = \text{front}$ to $i <= \text{rear}$
then display array [i];
- ⑤ In main function, create a object of class & call the above function .

Pseudocode:

```
1) class stack { int front; int rear;
    array[50];
    void Enqueue( char x )
    {
        if (rear == max-1)
            then queue overflow
        elseif (front == -1 & rear == -1)
            then rear = 0; front = 0
            array[rear] = x;
        else
            rear++ & array[rear] = x;
    }
```

void Dequeue()

```
{
    if (front == -1 & rear == -1) { Underflow condition }
    else if (front == rear) { head = 0; rear = 0; front = rear = -1 }
    else { front++ }.
```

void display

```
for (int i = front; i <= rear; i++)
{
```

```
cout << array[i] << endl;
```

```
}
```