



BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY

Munshi nagar, Andheri (W) ,Mumbai - 400058

DEPARTMENT OF MASTER OF COMPUTER APPLICATION

CLASS: F.Y. MCA

SEM: I

COURSE CODE: MC501

SUBJECT NAME: DATA STRUCTURES LAB

ROLL NO. : _2023510001____

BATCH: _D_

NAME: __VAIBHAV AGARWAL____

EXPERIMENT NO: 03

EXPERIMENT TITLE: Implement queue with insert, delete, traversal operations defined. The queue is an array representation. Use the same queue for device driver, where processes are waiting for usage of the device.

CODE:

```
#include <iostream>
#include <cstdio>
using namespace std;
#define size 5

class Queue
{
private:
    int front, rear;
    int arr[size];

public:
    Queue()
    {
        front = -1;
        rear = -1;
    }

    void enQueue(int n)
    {
        if (front == 0 && rear == size - 1)
        {
            cout << "\n Device queue is full. Process " << n << " cannot be enqueued.\n";
        }
        else
        {
```

```

        if (front == -1)
        {
            front = rear = 0;
            arr[rear] = n;
        }
        else
        {
            rear++;
            arr[rear] = n;
        }

        cout << "\n Process " << n << " enqueued for device access.\n";
    }
}

void deQueue()
{
    if ((front == -1) || (front == rear))
    {
        cout << " \n Device queue is empty. No process to dequeue.\n";
    }
    else
    {
        int temp = arr[front];

        if (front == rear)
        {
            front = rear = -1;
        }
        else
        {
            front++;
        }
        cout << "Process " << temp << " granted device access and dequeued.\n";
    }
}

void display()
{
    if ((front == -1) || (front == rear))
    {
        cout << " \n Device queue is empty.\n";
    }
    else
    {
        cout << "\n Device queue (front to rear): ";
        for (int i = front; i <= rear; i++)
            cout << arr[i] << " ";
    }

    cout << endl;
}

```

```
};

int main()
{
    Queue q;

    q.dequeue();
    q.display();

    q.enqueue(1);
    q.enqueue(2);
    q.enqueue(3);
    q.enqueue(4);
    q.enqueue(10);
    q.display();

    q.enqueue(6);

    q.dequeue();
    q.display();
}
```

OUTPUT:

Device queue is empty. No process to dequeue.

Device queue is empty.

Process 1 enqueued for device access.

Process 2 enqueued for device access.

Process 3 enqueued for device access.

Process 4 enqueued for device access.

Process 10 enqueued for device access.

Device queue (front to rear): 1 2 3 4 10

Device queue is full. Process 6 cannot be enqueued.
Process 1 granted device access and dequeued.

Device queue (front to rear): 2 3 4 10