

FAIZAN CHOUDHARY

20BCS021

DSA LAB

23<sup>rd</sup> November 2021

**CODE:** (code pasted in this format for readability)

```
#include <iostream>
using namespace std;

struct queue
{
    int info;
    struct queue *next;
};
struct queue *ptr, *front=NULL, *rear=NULL, *p;

int isEmpty ()
{
    if (front==NULL || rear==NULL)
        return 1;
    else
        return 0;
}

void display ()
{
    if (isEmpty()==1)
        cout<<"\nQueue is empty! Nothing to display\n";
    else
    {
        p=front;
        while (p!=rear)
        {
            cout<<p->info<<" <- ";
            p=p->next;
        }
        cout<<rear->info<<endl;
    }
}

int size ()
{
    if (isEmpty()==1)
        return 0;
    else
    {
        int count=1;
        for (p=front; p!=rear; p=p->next)
```

```

        count++;
        return count;
    }
}

void front_rear ()
{
    if (isEmpty()==1)
        cout<<"\nQueue is empty..."<<endl;
    else
    {
        cout<<"\nFront element is: "<<front->info;
        cout<<"\nRear element is: "<<rear->info<<endl;
    }
}

void enqueue (int n)
{
    ptr=(struct queue *) malloc (sizeof(struct queue));
    if (ptr==NULL)
    {
        cout<<"\nMemory could not be allocated!\n";
        return;
    }
    if (front==NULL)
    {
        front=rear=ptr;
    }
    else
    {
        rear->next=ptr;
        rear=ptr;
    }
    ptr->info=n;
    ptr->next=NULL;
    display();
}

void dequeue ()
{
    if (isEmpty()==1)
        cout<<"\nQueue Underflow! Stack is empty..."<<endl;
    else
    {
        cout<<"\nDequeuing front element: "<<front->info<<endl;
        front=front->next;
        display();
    }
}

int main()
{
    cout<<"\nFAIZAN CHOUDHARY\n20BCS021\n";
}

```

```

int ch,n;

while (true)
{
    A:
    cout<<"\nMENU:\n1. Enqueue\n2. Dequeue\n3. Display front and rear elements\n4.
Check if queue is empty\n5. Size of the queue\n6. Display queue\n7. Exit\n";
    cin>>ch;
    switch (ch)
    {
        case 1: cout<<"\nEnter the element to be enqueued: ";
                cin>>n;
                enqueue(n);
                break;
        case 2: dequeue();
                break;
        case 3: front_rear();
                break;
        case 4: if (isEmpty()==1)
                    cout<<"\nQueue is empty!\n";
                else
                    cout<<"\nQueue is not empty.\n";
                break;
        case 5: cout<<"\nSize of the queue is: "<<size()<<endl;
                break;
        case 6: cout<<"\nQueue elements: "<<endl;
                display();
                break;
        case 7: exit(0);
        default: cout<<"\nWrong choice! Enter again...\n";
                goto A;
    }
}
}

```

## OUTPUT:

FAIZAN CHOUDHARY  
20BCS021

MENU:

1. Enqueue
  2. Dequeue
  3. Display front and rear elements
  4. Check if queue is empty
  5. Size of the queue
  6. Display queue
  7. Exit
- 1

Enter the element to be enqueued: 22  
22

MENU:

1. Enqueue
  2. Dequeue
  3. Display front and rear elements
  4. Check if queue is empty
  5. Size of the queue
  6. Display queue
  7. Exit
- 1

Enter the element to be enqueued: 33  
22 <- 33

MENU:

1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if queue is empty
5. Size of the queue
6. Display queue
7. Exit

1

Enter the element to be enqueued: 44  
22 <- 33 <- 44

MENU:

1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if queue is empty
5. Size of the queue
6. Display queue
7. Exit

2

Dequeuing front element: 33  
44

MENU:

1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if queue is empty
5. Size of the queue
6. Display queue
7. Exit

3

Front element is: 22  
Rear element is: 44

MENU:

1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if queue is empty
5. Size of the queue
6. Display queue
7. Exit

2

Dequeuing front element: 44

Queue is empty! Nothing to display

MENU:

1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if queue is empty
5. Size of the queue
6. Display queue
7. Exit

5

Size of the queue is: 3

MENU:

1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if queue is empty
5. Size of the queue
6. Display queue
7. Exit

6

Queue elements:

Queue is empty! Nothing to display

MENU:

1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if queue is empty
5. Size of the queue
6. Display queue
7. Exit

2

Dequeuing front element: 22  
33 <- 44

MENU:

1. Enqueue
2. Dequeue
3. Display front and rear elements
4. Check if queue is empty
5. Size of the queue
6. Display queue
7. Exit

7