

**NATIONAL UNIVERSITY OF COMPUTER AND  
EMERGING SCIENCES  
PROGRAM: SOFTWARE ENGINEERING**



***DATA STRUCTURES LAB***  
**LAB TASK-07**

**SUBMITTED BY:**

**Name: Ahmed Ali**

**Roll No: 22P-9318**

**INSTRUCTOR NAME: Sir Saood Sarwar**  
**A DEPARTMENT OF COMPUTER SCIENCE**

## **Q1 CODE:**

```
//Front->dequeue and rear->enqueue

#include<iostream>

using namespace std;

class queue
{
    private:
        int size;
        int *arr;
        int front;
        int rear;
        bool dequeued_once;
        int last_dequeued_value;

    public:
        queue(int size)
        {
            this->size=size;
            arr=new(nothrow)int[size];
            front=-1;
            rear=-1;
            dequeued_once=false;
            last_dequeued_value=-1;
        }
        ~queue()
        {
            delete []arr;
```

```

}

bool is_empty()
{
    return(front==0);
}

bool is_full()
{
    return(rear==size-1);
}

void enqueue(int data)
{
    if(is_full())
    {
        cout<<"Queue is full"<<endl;
    }
    else
    {
        if(is_empty())
        {
            front=0;
        }
        arr[++rear]=data;
        cout<<"Enqueued: "<<data<<endl;
    }
}

void dequeue()
{
    if(is_empty())
    {

```

```

        cout<<"Queue is empty"<<endl;
    }
    else
    {
        last_dequeued_value=arr[front++];
        dequeued_once=true;
        cout<<"Dequeued successfully"<<endl;
        if(front>rear)
        {
            front=rear=-1;
        }
    }
}

void front_value()
{
    if(!dequeued_once)
    {
        cout<<"Nothing is dequeued so no front value is there."<<endl;
    }
    else if(is_empty())
    {
        cout<<"Queue is empty but last dequeued value:
"<<last_dequeued_value<<endl;
    }
    else
    {
        cout<<"Front value: "<<arr[front]<<endl;
    }
}

```

```
};
```

```
int main()
```

```
{
```

```
    int size,choice,value;
```

```
    cout<<"Enter the size of the queue: ";
```

```
    cin>>size;
```

```
    queue q(size);
```

```
    while(true)
```

```
    {
```

```
        cout<<endl<<"Menu:"<<endl;
```

```
        cout<<"1. Enqueue"<<endl;
```

```
        cout<<"2. Dequeue"<<endl;
```

```
        cout<<"3. Front value"<<endl;
```

```
        cout<<"4. Check if the queue is empty"<<endl;
```

```
        cout<<"5. Check if the queue is full"<<endl;
```

```
        cout<<"6. Exit"<<endl;
```

```
        cout<<"Enter your choice: ";
```

```
        cin>>choice;
```

```
        switch(choice)
```

```
        {
```

```
            case 1:
```

```
                cout<<"Enter value to enqueue: ";
```

```
                cin>>value;
```

```
                q.enqueue(value);
```

```
                break;
```

```
            case 2:
```

```
                q.dequeue();
```

```
break;
```

case 3:

```
q.front_value();
```

```
break;
```

case 4:

```
if(q.is_empty())
```

```
{
```

```
    cout<<"The queue is empty."<<endl;
```

```
}
```

```
else
```

```
{
```

```
    cout<<"The queue is not empty."<<endl;
```

```
}
```

```
break;
```

case 5:

```
if(q.is_full())
```

```
{
```

```
    cout<<"The queue is full."<<endl;
```

```
}
```

```
else
```

```
{
```

```
    cout<<"The queue is not full."<<endl;
```

```
}
```

```
break;
```

case 6:

```

        cout<<"Exiting..."<<endl;

        return 0;

```

default:

```

        cout<<"Invalid choice. Please try again."<<endl;

```

```

    }

}

return 0;

}

```

## **Output-01:**

D:\SUMMER'24\Data Structures LAB\LAB TASK 7\Q1\_Ahmed\_9318.exe

```

Enter the size of the queue: 3

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 1
Enter value to enqueue: 10
Enqueued: 10

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 3
Nothing is dequeued so no front value is there.

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 2
Dequeued successfully

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 3
Queue is empty but last dequeued value: 10

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 1
Enter value to enqueue: 1

```

**01**

```

Enqueued: 1

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 1
Enter value to enqueue: 2
Enqueued: 2

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 1
Enter value to enqueue: 3
Enqueued: 3

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 3
Front value: 1

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 2
Dequeued successfully

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 3

```

**02**

D:\SUMMER'24\Data Structures LAB\LAB TASK 7\Q1\_Ahmed\_9318.exe

```

Front value: 2

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 4
The queue is not empty.

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 5
The queue is full.

Menu:
1. Enqueue
2. Dequeue
3. Front value
4. Check if the queue is empty
5. Check if the queue is full
6. Exit
Enter your choice: 6
Exiting...

-----
Process exited after 88.12 seconds with return value 0
Press any key to continue . . .

```

**03**

## **Q2 CODE:**

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
class node
```

```
{
```

```
    public:
```

```
        string info;
```

```
        node *next;
```

```
        node(string info)
```

```
        {
```

```
            this->info=info;
```

```
            next=nullptr;
```

```
        }
```

```
};
```

```
class queue
```

```
{
```

```
    private:
```

```
        node *front;
```

```
        node *rear;
```

```
    public:
```

```
        queue()
```

```
        {
```

```
            front=nullptr;
```

```
            rear=nullptr;
```



```
}
```

```
bool empty()
```

```
{
```

```
    if(front==nullptr)
```

```
    {
```

```
        return true;
```

```
    }
```

```
    else
```

```
    {
```

```
        return false;
```

```
    }
```

```
}
```

```
void add(string info)
```

```
{
```

```
    node *newnode=new node(info);
```

```
    if(empty())
```

```
    {
```

```
        front=rear=newnode;
```

```
    }
```

```
    else
```

```
    {
```

```
        rear->next=newnode;
```

```
        rear=newnode;
```

```
    }
```

```
}
```

```
string process()
```

```

    {
        if(empty())
        {
            cout<<"Empty queue"<<endl;
            return "";
//            return "empty queue";
        }
        node *temp=front;
        string info=front->info;
        front=front->next;

        if(front==nullptr)
        {
            rear=nullptr;
        }
        delete temp;
        return info;
    }

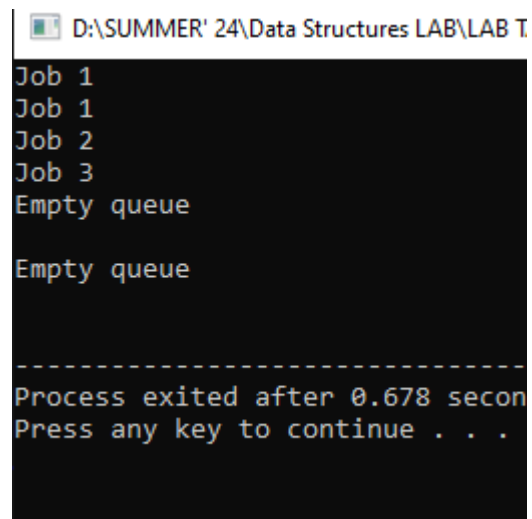
string current_info()
{
    if(empty())
    {
        cout<<"Empty queue"<<endl;
        return "";
//        return "empty queue";
    }
    return front->info;
}

```

```
};

int main()
{
    queue pq;
    pq.add("Job 1");
    pq.add("Job 2");
    pq.add("Job 3");
    cout<<pq.current_info()<<endl;
    cout<<pq.process()<<endl;
    cout<<pq.process()<<endl;
    cout<<pq.process()<<endl;
    cout<<pq.process()<<endl;
    cout<<pq.current_info()<<endl;
    return 0;
}
```

## **Output-02:**



```
D:\SUMMER' 24\Data Structures LAB\LAB T...
Job 1
Job 1
Job 2
Job 3
Empty queue
Empty queue

-----
Process exited after 0.678 seconds
Press any key to continue . . .
```

---

