```
//QUEUE ADT
#include <iostream>
using namespace std;

class Queue{
    private:
    int front,rear;
    int arr[10];

    public:
    Queue(){
        front=0;
        rear=0;
    }
    bool isEmpty(){
        if(rear==front){
            return true;
        }
        else{
            return false;
        }
    }

    bool isFull(){
        if((rear+1)%10==front){
            return true;
        }
        else{
            return false;
        }
    }

    void enqueue(int element){
        if(isFull()){
            cout<<"Queue is full."<<endl;
        }
        else{
            arr[rear]=element;
            rear=((rear+1)%10);
        }
    }

    int dequeue(){
        int x;
        if(isEmpty()){
            cout<<"Queue is empty."<<endl;
            return 0;
        }
        else{
            x=arr[front];
            front=((front+1)%10);
            return x;
        }
    }

    void display(){
        if(front<rear){
            for(int i=front;i<rear;i++){
                cout<<arr[i]<<"   ";
```

```
            }
        }
        else{
            for(int i=front;i<10;i++){
                cout<<arr[i]<<"   ";
            }
            for(int i=0;i<rear;i++){
                cout<<arr[i]<<"   ";
            }
        }
        cout<<endl;
        cout<<"Front: "<<front<<endl;
        cout<<"Rear: "<<rear<<endl;
        cout<<endl;
    }

    int menu(){
        int ch;
        cout<<endl;
        cout<<"1. Enqueue"<<endl;
        cout<<"2. Dequeue"<<endl;
        cout<<"3. Display all elements"<<endl;
        cout<<"4. Exit"<<endl;
        cout<<"Enter the operation to be performed"<<endl;
        cin>>ch;
        return ch;
    }
};
```

```
#include <iostream>
#include
<C:\Users\Shounak\Desktop\DSAL\Assignment3_CircularQueue\Cqueue.cpp>
using namespace std;

int main(){
    Queue q;
   while(1){
        int x=q.menu();
        switch(x){
            case 1:
                cout<<"enter element to be enqueued."<<endl;
                int elem;
                cin>>elem;
                q.enqueue(elem);
                break;

            case 2:
                cout<<"Dequeued element is: "<<q.dequeue()<<endl;
                break;

            case 3:
                cout<<endl;
                cout<<"Queue"<<endl;
                q.display();
                break;

            case 4:
```

```
                exit(0);

            }
        }
        return 0;
}
```

```
1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
1

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
2

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
3

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
4

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
5

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
6

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
```

```
enter element to be enqueued.
7

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
8

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
9

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
10
Queue is full.

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
3

Queue
1   2   3   4   5   6   7   8   9
Front: 0
Rear: 9


1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
2
Dequeued element is: 1

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
2
Dequeued element is: 2
```

```
1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
2
Dequeued element is: 3

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
2
Dequeued element is: 4

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
2
Dequeued element is: 5

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
3

Queue
6   7   8   9
Front: 5
Rear: 9


1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
10

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
11

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
```

```
1
enter element to be enqueued.
12

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
13

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
14

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
1
enter element to be enqueued.
15
Queue is full.

1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
3

Queue
6   7   8   9   10   11   12   13   14
Front: 5
Rear: 4


1. Enqueue
2. Dequeue
3. Display all elements
4. Exit
Enter the operation to be performed
4

Process returned 0 (0x0)   execution time : 38.783 s
Press any key to continue.
```