

Question3.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  template <class T>
5  class Queue
6  {
7  private:
8      int front, rear;
9      unsigned capacity;
10     T *array;
11
12 public:
13     Queue(int capacity)
14     {
15         array = new T[capacity];
16         front = -1;
17         rear = -1;
18         this->capacity = capacity;
19     }
20
21     bool isEmpty()
22     {
23         return (rear == -1 && front == -1);
24     }
25
26     bool isFull()
27     {
28         return ((rear + 1) % capacity == front);
29     }
30
31     void enqueue(T element)
32     {
33         if (isFull())
34         {
35             cerr << "Queue is full cannot enqueue." << endl;
36             return;
37         }
38         else if (isEmpty())
39         {
40             front = rear = 0;
41         }
42         else
43         {
44             rear = (rear + 1) % capacity;
45         }
46         array[rear] = element;
47         cout << array[rear] << " is enqueued in the queue." << endl;
48     }
49
50     void dequeue()
51     {
52         if (isEmpty())
53         {
```

```
54         cerr << "Queue is Empty cannot dequeue." << endl;
55         return;
56     }
57     else if (front == rear)
58     {
59         cout << array[front] << " is dequeued from the queue." << endl;
60         rear = front = -1;
61     }
62     else
63     {
64         cout << array[front] << " is dequeued from the queue." << endl;
65         front = (front + 1) % capacity;
66     }
67 }
68
69 T peek()
70 {
71     if (isEmpty())
72     {
73         cerr << "Queue is empty." << endl;
74     }
75     return array[front];
76 }
77
78 int queue_size()
79 {
80     if (isEmpty())
81         return 0;
82     else if (rear >= front)
83     {
84         return rear - front + 1;
85     }
86     else
87     {
88         return (capacity - front + rear) % capacity;
89     }
90 }
91 };
92
93 int main()
94 {
95     Queue<char> q(10);
96     q.enqueue('a');
97     q.enqueue('b');
98     q.enqueue('c');
99     q.enqueue('d');
100    q.enqueue('e');
101    q.enqueue('f');
102    q.enqueue('g');
103    q.enqueue('h');
104    q.enqueue('h');
105    q.enqueue('h');
106    q.enqueue('h');
107    q.enqueue('h');
108    q.enqueue('h');
109    q.dequeue();
```

```
110     q.dequeue();
111     cout << "Front element " << q.peek() << endl;
112     cout << "Size of queue " << q.queue_size() << endl;
113     return 0;
114 }
115
```