Question03.cpp

```cpp
// <----Lab 06 - Queues---->

// Q3. Give the C++ code to implement the Generic Queue using array. Please change the
//     private data members as you are now using an array.


#include <iostream>
using namespace std;

template <class T>
class Queue
{
private:
    int front, rear;
    unsigned capacity;
    T *array;

public:
    Queue(int capacity)
    {
        array = new T[capacity];
        front = -1;
        rear = -1;
        this->capacity = capacity;
    }

    bool isEmpty()
    {
        return (rear == -1 && front == -1);
    }

    bool isFull()
    {
        return ((rear + 1) % capacity == front);
    }

    void enqueue(T element)
    {
        if (isFull())
        {
            cerr << "Queue is full cannot enqueue." << endl;
            return;
        }
        else if (isEmpty())
        {
            front = rear = 0;
        }
        else
        {
            rear = (rear + 1) % capacity;
        }
        array[rear] = element;
        cout << array[rear] << " is enqueued in the queue." << endl;
```

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

```
110     q.enqueue('h');
111     q.enqueue('h');
112     q.enqueue('h');
113     q.enqueue('h');
114     q.enqueue('h');
115     q.dequeue();
116     q.dequeue();
117     cout << "Front element " << q.peek() << endl;
118     cout << "Size of queue " << q.queue_size() << endl;
119     return 0;
120 }
```