Name : Farhan Ahmad

Sap id : 56193

Course : DSA LAB

Lab Task 5

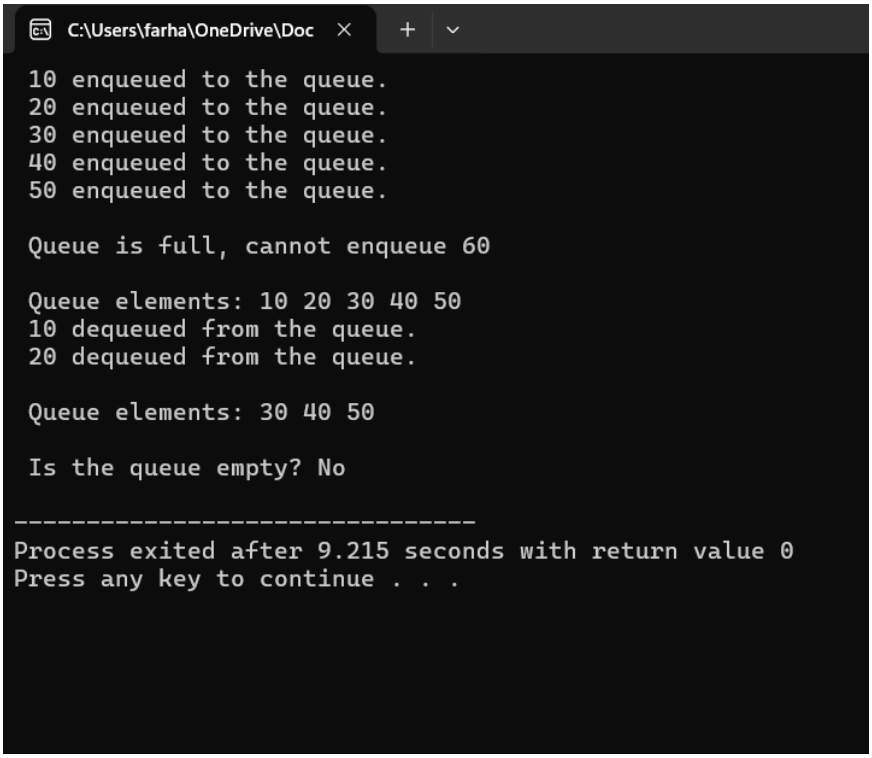
Qno # 1

```
#include <iostream>
using namespace std;
class Queue
{
private:
   int front, rear;
   int size;
   int* arr;
public:

   Queue()
   {
      size = 5;
      front = rear = -1;
      arr = new int[size];
   }
   void enqueue(int value)
   {
```

```
    if (rear == size - 1)
    {
        cout << "\n Queue is full, cannot enqueue " << value << endl;
        return;
    }
    if (front == -1)
    {
        front = 0;
    }
    arr[++rear] = value;
    cout << " " << value << " enqueued to the queue." << endl;
}

void dequeue()
{
    if (isEmpty())
    {
        cout << "\n Queue is empty, cannot dequeue." << endl;
        return;
    }
    cout << " " << arr[front] << " dequeued from the queue." << endl;
    front++;
    if (front > rear)
    {
```

```
            front = rear = -1;
        }
    }
    bool isEmpty()
    {
        return (front == -1);
    }
    void display()
    {
        if (isEmpty())
        {
            cout << " Queue is empty." << endl;
            return;
        }
        cout << "\n Queue elements: ";
        for (int i = front; i <= rear; i++)
        {
            cout << arr[i] << " ";
        }
        cout << endl;
    }
    ~Queue()
    {
delete[] arr;
}
```

```
};
int main()
{

Queue q;
q.enqueue(10);
q.enqueue(20);
q.enqueue(30);
q.enqueue(40);
q.enqueue(50);
q.enqueue(60);
q.display();
q.dequeue();
q.dequeue();
q.display();
cout << "\n Is the queue empty? " << (q.isEmpty() ? "Yes" : "No") << endl;
return 0;
}
```



Qno # 2

#include <iostream>

#include <string>

using namespace std;

#define MAX 100

class Queue

{

char arr[MAX];

int front, rear;

public:

Queue()

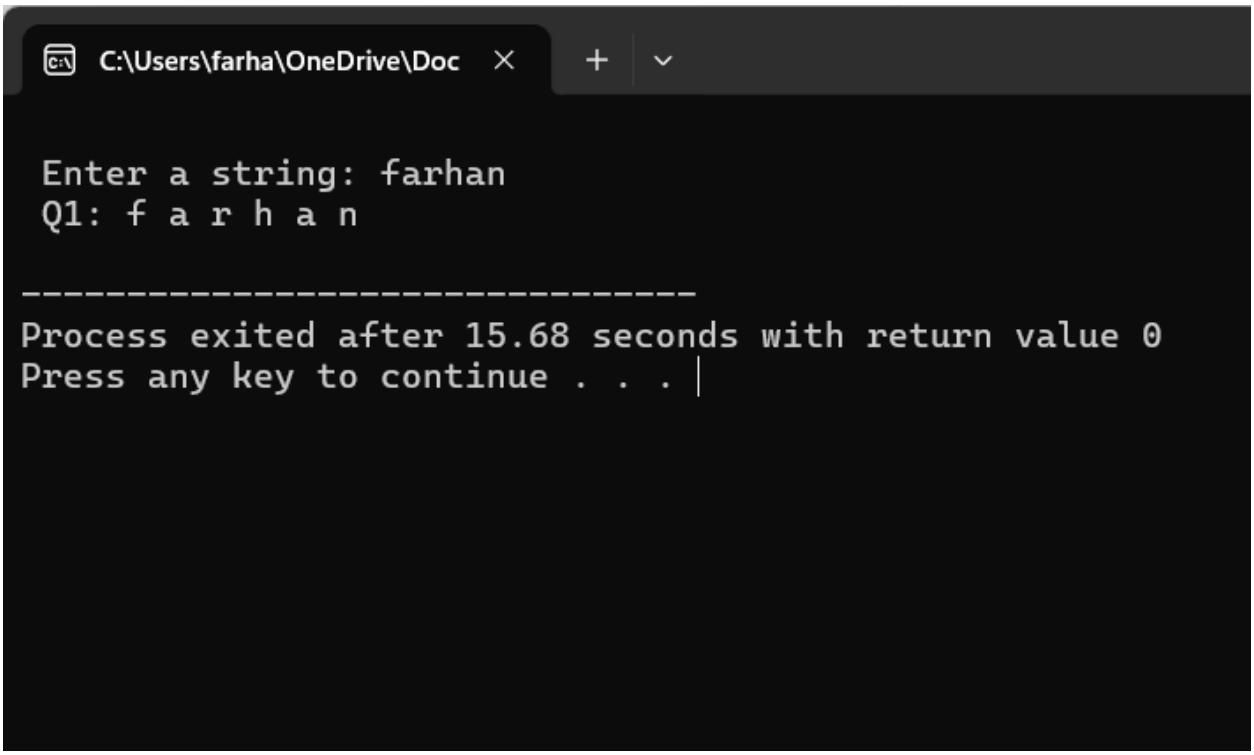
{

```
    front = -1;
    rear = -1;
}
bool isEmpty()
{
    return front == -1 || front > rear;
}

void enqueue(char value)
{
    if (rear == MAX - 1)
    {
        cout << " Queue is full!" << endl;
    }
    else
    {
        if (front == -1) front = 0;
        arr[++rear] = value;
    }
}
char dequeue()
{
    if (isEmpty())
    {
        cout << " Queue is empty!" << endl;
```

```
        return '\0';
    }
    return arr[front++];
}
void display()
{
    if (isEmpty())
    {
        cout << " Queue is empty" << endl;
        return;
    }
    for (int i = front; i <= rear; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

void concatenate(Queue& q)
{
    while (!q.isEmpty())
    {
        enqueue(q.dequeue());
    }
}
```

```
};
int main()
{
    string input;
    cout << "\n Enter a string: ";
    getline(cin, input);
    Queue queues[10];
    int queueIndex = 0;
    Queue currentQueue;
    for (int i = 0; i < input.length(); i++)
    {
        if (input[i] != ' ')
        {
            currentQueue.enqueue(input[i]);
        }
        else
        {
            if (!currentQueue.isEmpty())
            {
                queues[queueIndex++] = currentQueue;
                currentQueue = Queue();
            }
        }
    }
```

```
    if (!currentQueue.isEmpty())
    {
        queues[queueIndex++] = currentQueue;
    }

    for (int i = 0; i < queueIndex; i++)
    {
        cout << " Q" << i + 1 << ": ";
        queues[i].display();
}
Queue resultQueue = queues[0];
for (int i = 1; i < queueIndex; i++)
{

resultQueue.concatenate(queues[i]);
cout << "\n Concatenated Queue: ";
resultQueue.display();
}
return 0;
}
```


C:\Users\farha\OneDrive\Doc
Enter a string: farhan
Q1: f a r h a n

Process exited after 15.68 seconds with return value 0
Press any key to continue . . .