Page :
Date : / /

Assignment 11

Class-SE IV ROll NO-21430.
Batch - F4
D.O.S - 3/12/2020

Problem Statement-

Queues are frequently used in computer programming and a typical example is the creation of a job queue by on operating system. If the operating system does not use priorities then the jobs are produced in the order they enter the system. While c++ program for simulating job queue. Write functions to add job and delete job from queue

Learning objectives

To understand various concepts of Data structure queue to perform different operations.

Learning outcomes

Students will be able to perform different operations on queue.

S/W and H/W requirement.

Open source c++ tool.

Open source C++ IDE Eclipse

open source Linux.

Theory.

Queue

It is kind of list where items are inserted at one end & deleted from other end. Queue is a FIFO (first in first out)

Various features of OS are implemented using Queue

a. Scheduling Process

b. spooling

A queue of client process waiting to resume survice from the server process various applications software using non-linear data structure tree or graph receives a queue for BFS.

Different operations on queue using array are:

i> Initialize() - Inserts a queue by setting the value of rear and front to -1.

ii> Enque(). Inserts element at rear end of the queue.

iii> Dequeue() - Deletes front element and returns the same element.

iv> Empty(). It returns true if queue is empty else returns false.

v> Full() - It returns true if queue is full and retu rs false.

vi> print() - prints whole queue.

* Pseudocode -:

* ADT for class Node -:

```
class Node
{
    int jobno          // Job number
    Node * next        // next pointer.
    initialize()  // Initialize data member

};
```

* ADT for class Queue

```
class Queue
{
    Node * front
    Node * rear
    initialize()    // Initializes data member.
    Enqueue()       // Inserts node at rear.
    Dequeue()       // Deletes node at front.
    Empty()         // Checks if queue is full or empty
}
```

Pseudocode for Insert Job.

```
1. if( size>10)
   {
       print "Queue is full"
       return
   }
```

```
2. Size = size + 1
3. If ( rear==NULL)
   {
      QNode * newJob = new QNode ;
      nejob -> data = id
      newjob -> next = NULL
     front = rear = newJob
      return ;
   }
4. QNode * newJob = new QNode ;
5. NewJob -> data = id ;
6. NewJob -> next = NULL ;
rear -> next = newJob
8. rear = newJob

}
```

Pseudocode for Display

```
1. QNode * temp = front
2. While ( temp != NULL)
 {
    print temp -> data
    set temp = temp -> next

}
```

Psuedocode for DeleteJob

```
1. If ( front ==NULL)
   { print Queue is Empty
     return
   }
```

Complexity

```
2. size = size - 1
3. QNode * temp = front
4. front = front->next;
5. if( front==NULL)
   {
        rear=NULL;
   }
6. delete QNode.
7. END
```

Test Cases:

NO.	Description	Input	Actual o/p	Real o/p	Result
1.	Menu: 1.Insert 2.Show 3.Delete.	ch = 1 No.Job=3 Job=25 Job=36 Job=45	Job added	Job added	Pass.
2.	Menu: 1.Insert 2.Show 3.Delete	ch=2	25 36 45	25 36 45	Pass.

* Complexities

n = Number of Jobs in Queue

1. Enqueue -

Time Complexity = O(1)

2. Dequeue -

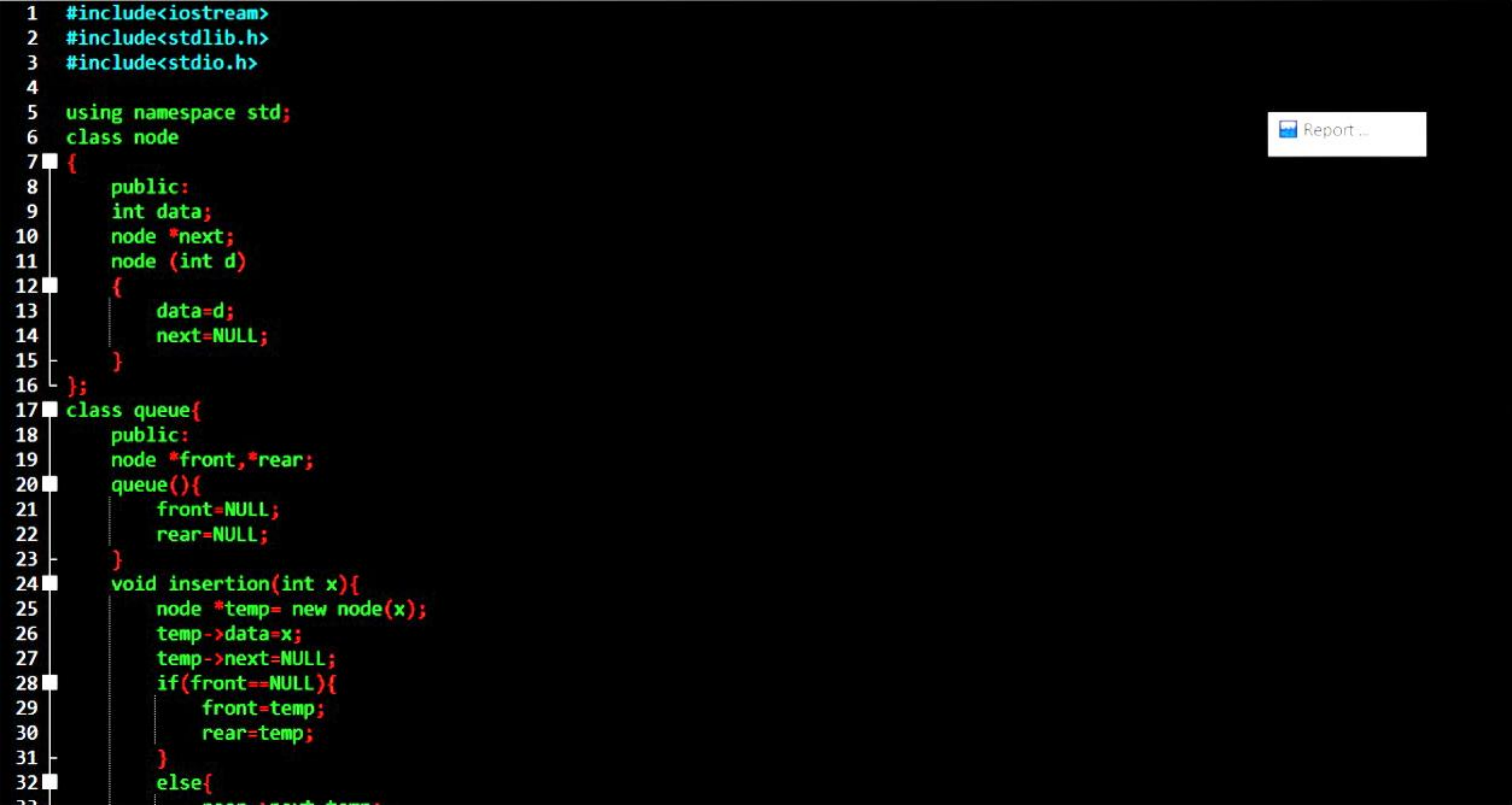
Time Complexity = O(1)

3. Display

Time complexity = O(n)

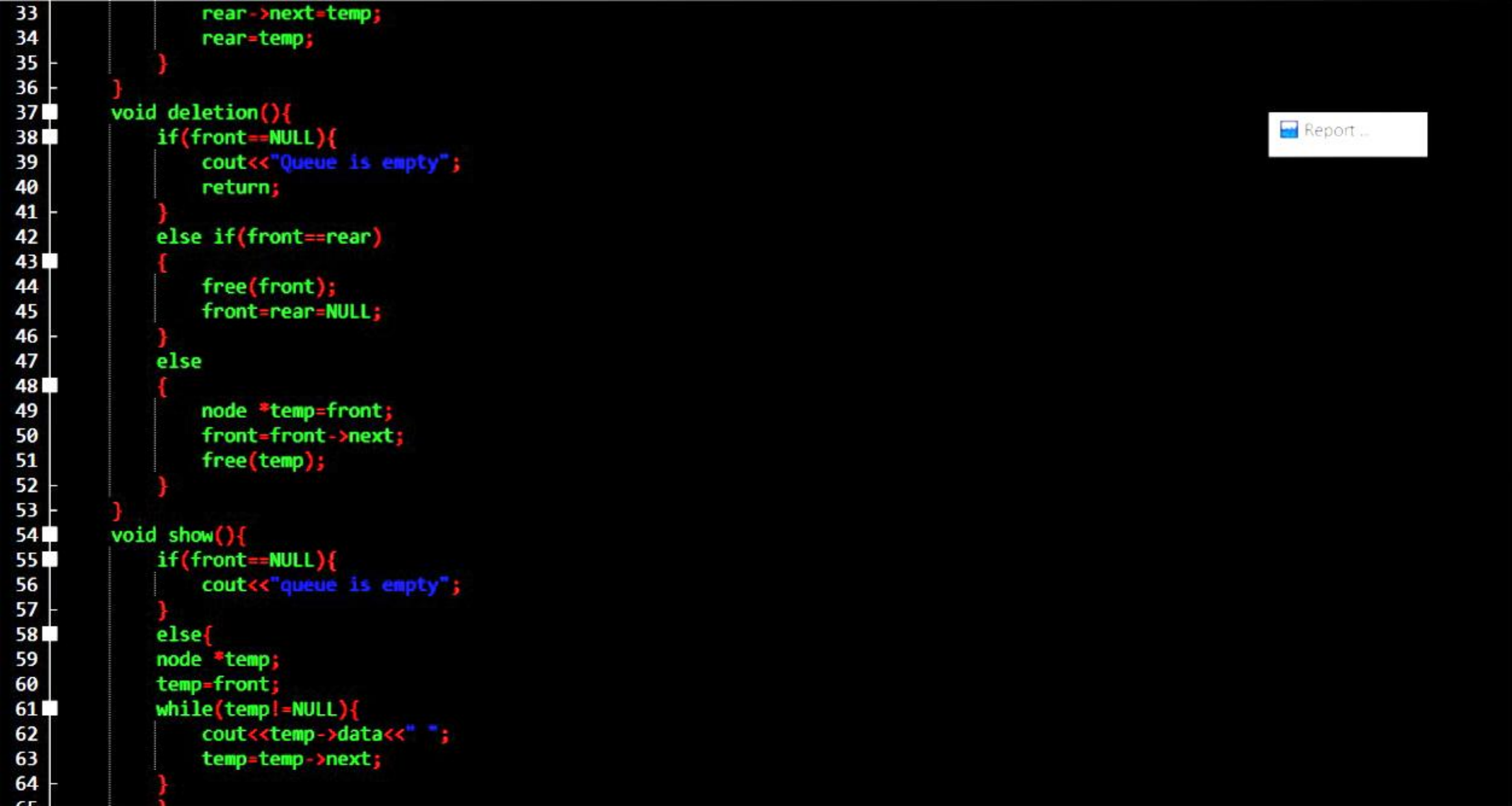
* Conclusion-:

We learnt to implement Queue data structure succesfully.

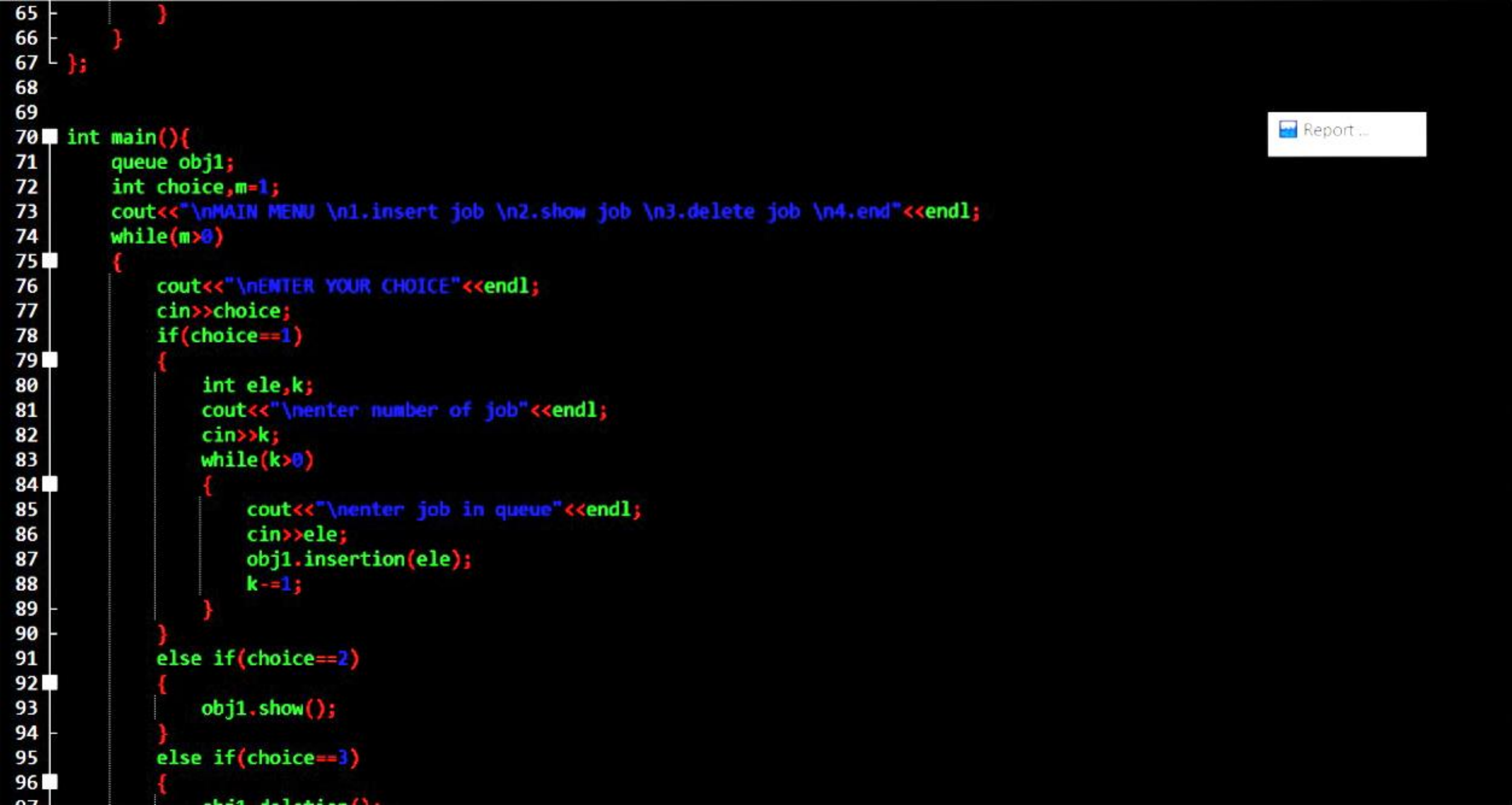


```
#include<iostream>
#include<stdlib.h>
#include<stdio.h>

using namespace std;
class node
{
    public:
    int data;
    node *next;
    node (int d)
    {
        data=d;
        next=NULL;
    }
};
class queue{
    public:
    node *front,*rear;
    queue(){
        front=NULL;
        rear=NULL;
    }
    void insertion(int x){
        node *temp= new node(x);
        temp->data=x;
        temp->next=NULL;
        if(front==NULL){
            front=temp;
            rear=temp;
        }
        else{
            rear->next=temp;
```

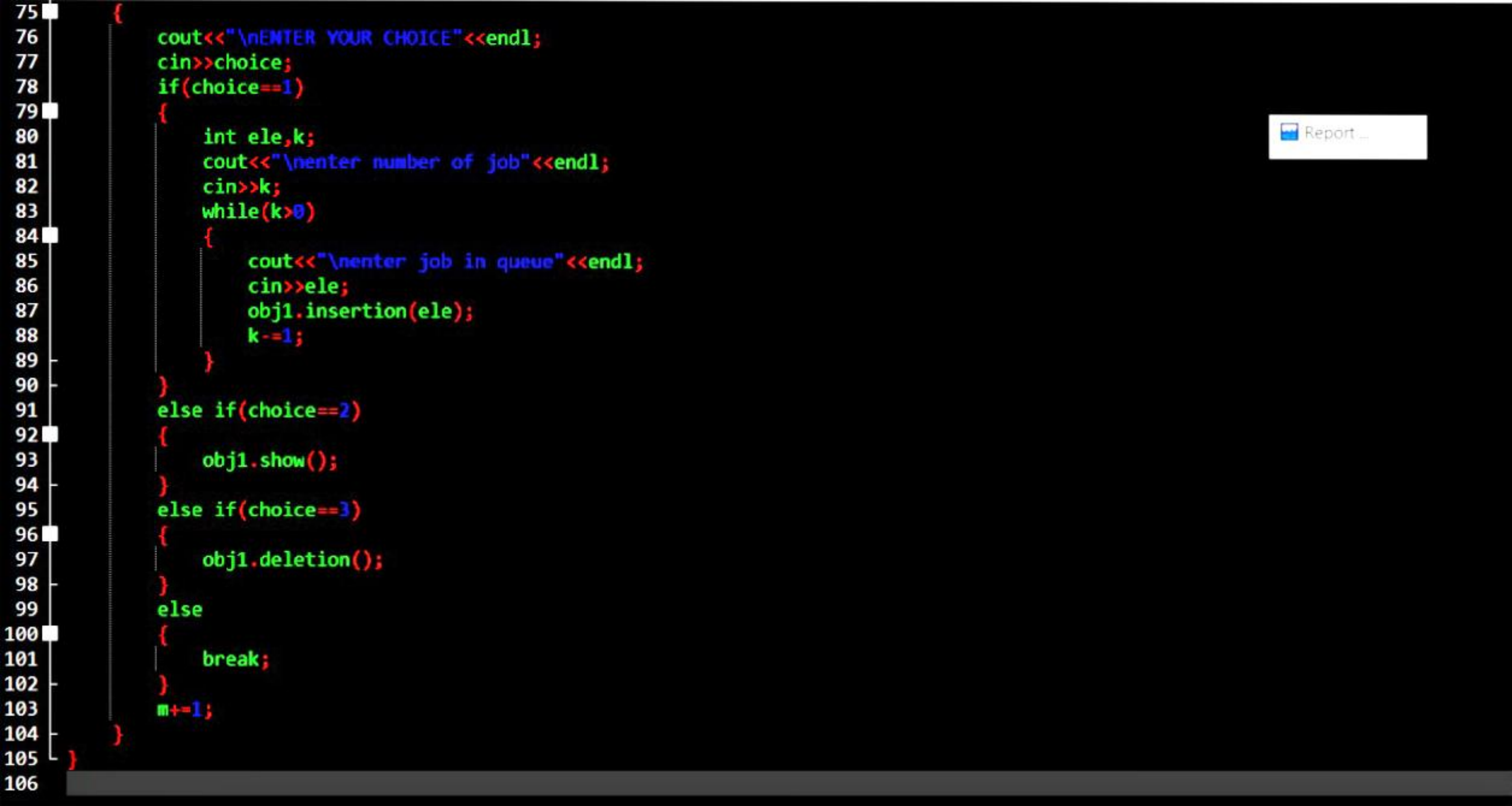


```
            rear->next=temp;
            rear=temp;
        }
    }
    void deletion(){
        if(front==NULL){
            cout<<"Queue is empty";
            return;
        }
        else if(front==rear)
        {
            free(front);
            front=rear=NULL;
        }
        else
        {
            node *temp=front;
            front=front->next;
            free(temp);
        }
    }
    void show(){
        if(front==NULL){
            cout<<"queue is empty";
        }
        else{
        node *temp;
        temp=front;
        while(temp!=NULL){
            cout<<temp->data<<" ";
            temp=temp->next;
        }
        }
```

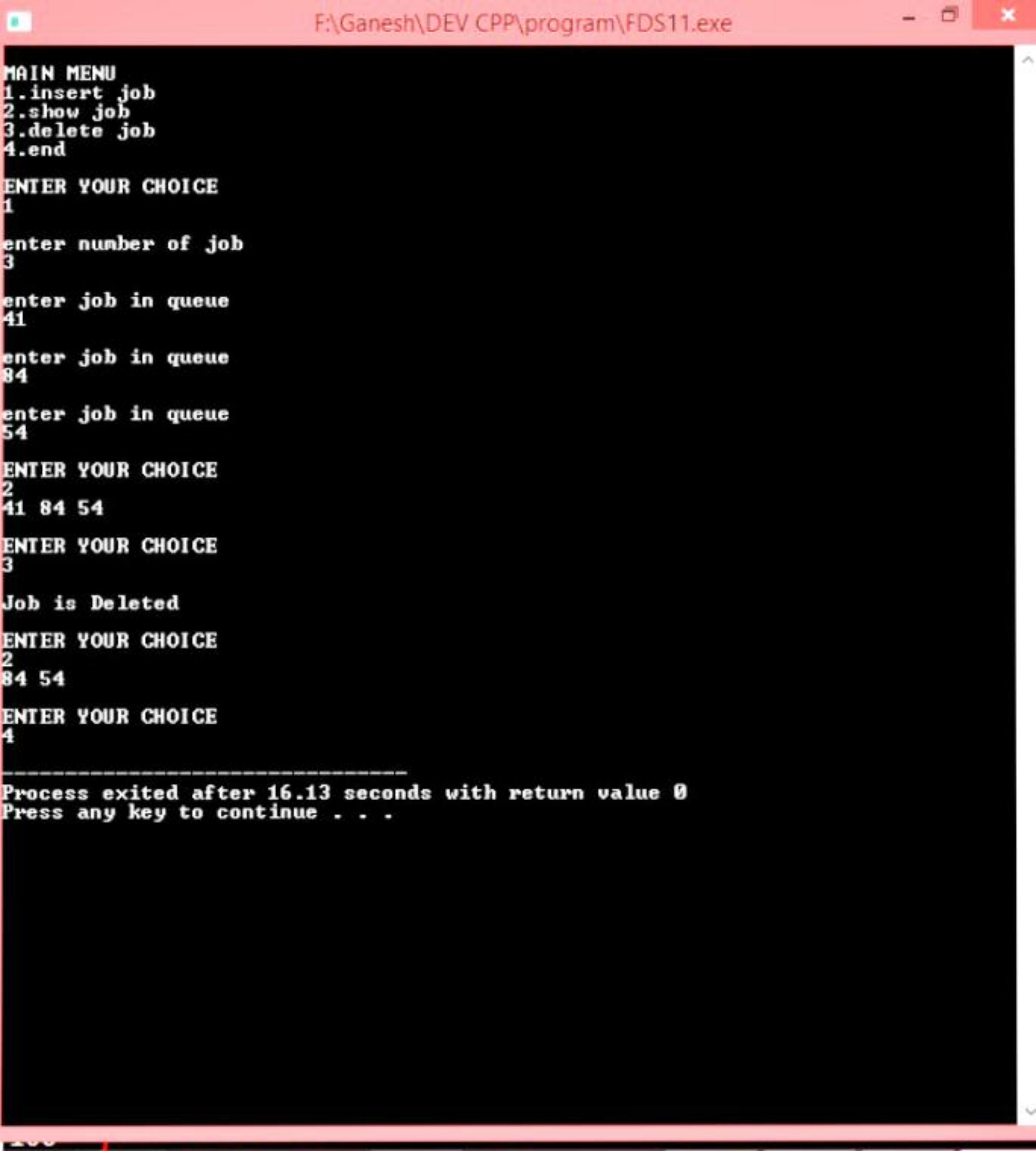


```
        }
    }
};


int main(){
    queue obj1;
    int choice,m=1;
    cout<<"\nMAIN MENU \n1.insert job \n2.show job \n3.delete job \n4.end"<<endl;
    while(m>0)
    {
        cout<<"\nENTER YOUR CHOICE"<<endl;
        cin>>choice;
        if(choice==1)
        {
            int ele,k;
            cout<<"\nenter number of job"<<endl;
            cin>>k;
            while(k>0)
            {
                cout<<"\nenter job in queue"<<endl;
                cin>>ele;
                obj1.insertion(ele);
                k-=1;
            }
        }
        else if(choice==2)
        {
            obj1.show();
        }
        else if(choice==3)
        {
            obj1.deletion();
```



```
    {
        cout<<"\nENTER YOUR CHOICE"<<endl;
        cin>>choice;
        if(choice==1)
        {
            int ele,k;
            cout<<"\nenter number of job"<<endl;
            cin>>k;
            while(k>0)
            {
                cout<<"\nenter job in queue"<<endl;
                cin>>ele;
                obj1.insertion(ele);
                k-=1;
            }
        }
        else if(choice==2)
        {
            obj1.show();
        }
        else if(choice==3)
        {
            obj1.deletion();
        }
        else
        {
            break;
        }
        m+=1;
    }
}
```



F:\Ganesh\DEV CPP\program\FDS11.exe

```
MAIN MENU
1.insert job
2.show job
3.delete job
4.end

ENTER YOUR CHOICE
1

enter number of job
3

enter job in queue
41

enter job in queue
84

enter job in queue
54

ENTER YOUR CHOICE
2
41 84 54

ENTER YOUR CHOICE
3

Job is Deleted

ENTER YOUR CHOICE
2
84 54

ENTER YOUR CHOICE
4

--------------------------------
Process exited after 16.13 seconds with return value 0
Press any key to continue . . .
```