queue\_07.cpp

**Compile:** g++ queue\_07.cpp -o queue\_07

**Run:** ./queue\_07

**Program:**

#include<iostream>

#define max 100

using namespace std;

int rear=-1;

int front=0;

int flag=0;

class queue

{

public:

intarr[max], a, size;

voidgetdata()

{

cout<<"Enter size of array: ";

cin>>size;

cout<<endl;

}

void insert()

{

if(rear==size-1)

{

cout<<"Queue Overflow..."<<endl;

}

else

{

++rear;

cout<<"Insert an element: ";

cin>>a;

arr[rear]=a;

cout<<endl;

}

}

void remove()

{

if(front>rear)

{

cout<<"Queue Underflow..."<<endl;

flag=1;

}

else

{

cout<<arr[front]<<" deleted from queue..."<<endl;

front++;

cout<<"Rear= "<<rear<<endl;

cout<<"Front= "<<front<<endl;

}

}

void display()

{

if(flag==1)

{

cout<<"Queue is empty..."<<endl;

}

else

{

cout<<"Elements in the array are: ";

for(inti=front; i<=rear; i++)

{

cout<<arr[i]<<" ";

}

cout<<endl;

}

}

};

int main()

{

queue q;

q.getdata();

int n;

n=0;

do

{

cout<<"Queue operation...\n1)Insert\n2)Delete\n3)Display\n4)Exit\nSelect an operation: ";

cin>>n;

switch(n)

{

case 1:

q.insert();

break;

case 2:

q.remove();

break;

case 3:

q.display();

break;

case 4:

cout<<"Exited..."<<endl;

default:

cout<<"Invalid Input..."<<endl;

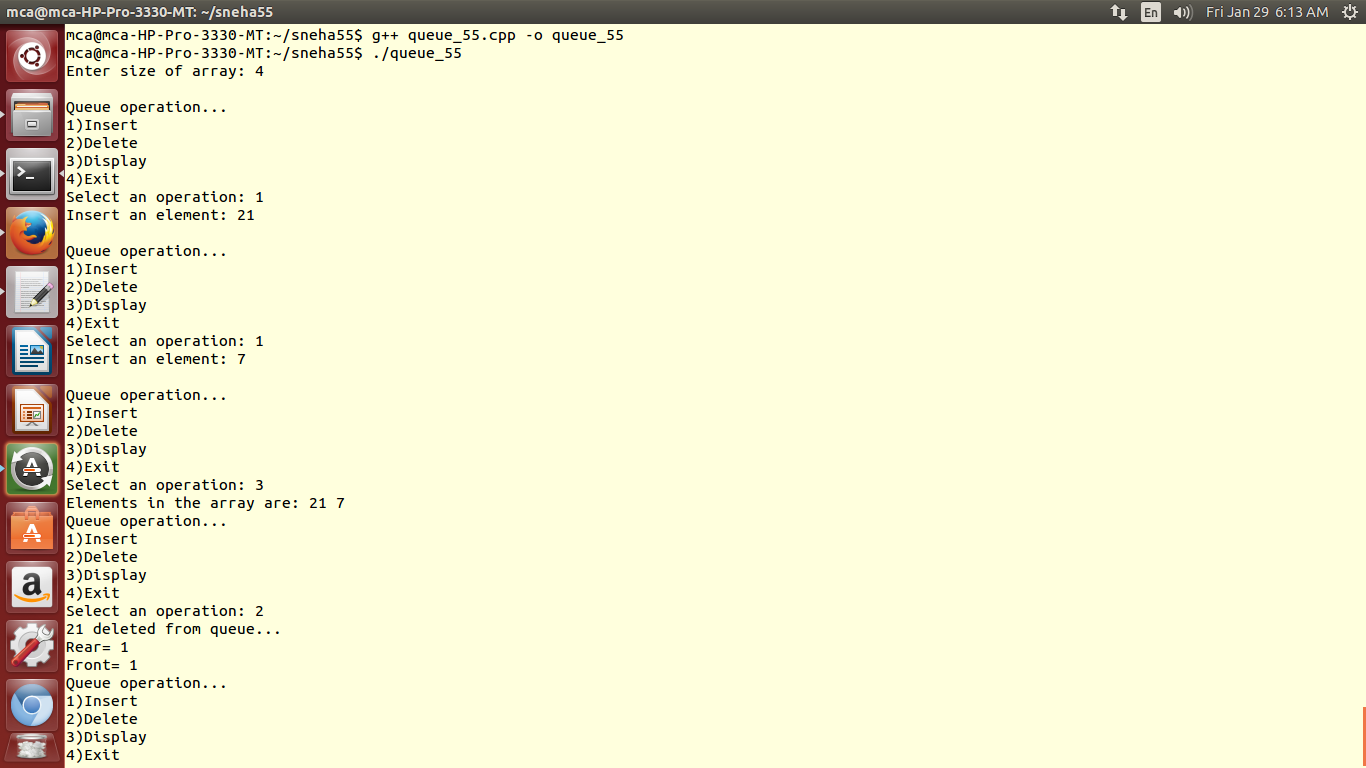
}

}while(n!=4);

return 0;

}

**Output:**

****