Program:

*#include* <iostream>

*#define* MAX 10

*using* *namespace* std;

*struct* queue

{

*int* data[MAX];

*int* front, rear;

};

*class* Queue

{

*struct* queue q;

*public:*

    Queue() { q.front *=* q.rear *=* *-*1; }

*int* IsEmpty();

*int* IsFull();

*void* Enqueue(*int*);

*int* Delqueue();

*void* Display();

};

*int* Queue::IsEmpty()

{

*return* (q.front *==* q.rear) *?* 1 *:* 0;

}

*int* Queue::IsFull()

{

*return* (q.rear *==* MAX *-* 1) *?* 1 *:* 0;

}

*void* Queue::Enqueue(*int* *x*)

{

    q.data[*++*q.rear] *=* *x*;

}

*int* Queue::Delqueue()

{

*return* q.data[*++*q.front];

}

*void* Queue::Display()

{

*int* i;

    cout *<<* "\n";

*for* (i *=* q.front *+* 1; i *<=* q.rear; i*++*)

        cout *<<* q.data[i] *<<* ", ";

}

*int* main()

{

    Queue obj;

*int* ch, x;

*do*

    {

        cout *<<* "\n\* \* \* \* \* \* \* \* \*";

        cout *<<* "\n\* 1.Insert Job  \*";

        cout *<<* "\n\* 2.Delete Job  \*";

        cout *<<* "\n\* 3.Display     \*";

        cout *<<* "\n\* 4.Exit        \*";

        cout *<<* "\n\* \* \* \* \* \* \* \* \*";

        cout *<<* "\nEnter your choice: ";

        cin *>>* ch;

*switch* (ch)

        {

*case* 1:

*if* (*!*obj.IsFull())

            {

                cout *<<* "\n Enter Data: ";

                cin *>>* x;

                obj.Enqueue(x);

            }

*else*

                cout *<<* "Queue is overflow";

*break*;

*case* 2:

*if* (*!*obj.IsEmpty())

                cout *<<* "\n Deleted Element= " *<<* obj.Delqueue();

*else*

            {

                cout *<<* "\n Queue is underflow";

            }

            cout *<<* "\nRemaining jobs :";

            obj.Display();

*break*;

*case* 3:

*if* (*!*obj.IsEmpty())

            {

                cout *<<* "\n Queue contains:";

                obj.Display();

            }

*else*

                cout *<<* "\n Queue is empty";

*break*;

*case* 4:

            cout *<<* "\n Exit";

        }

    } *while* (ch *!=* 4);

*return* 0;

}

Output :



