Name : Aditi Sant , AI&DS-B1 batch , Roll no.: 23211

**EXPERIMENT NO:11**

#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;

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

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

cout << "\n";

}

int main()

{

cout << "Name : Aditi Sant , AI&DS-B1 batch , Roll no.: 23211" << endl;

Queue obj;

int ch, x;

do

{

cout << "\n1.Insert Job\n2.Delete Job\n3.Display\n4.Exit\nEnter your choice : ";

cin >> ch;

switch (ch)

{

case 1:

if (!obj.isfull())

{

cout << "\nEnter data : ";

cin >> x;

obj.enqueue(x);

}

else

cout << "\nQueue is overflow!!!\n";

break;

case 2:

if (!obj.isempty())

cout << "\nDeleted Element = " << obj.delqueue() << endl;

else

{

cout << "\nQueue is underflow!!!\n";

}

cout << "\nRemaining Jobs : \n";

obj.display();

break;

case 3:

if (!obj.isempty())

{

cout << "\nQueue contains : \n";

obj.display();

}

else

cout << "\nQueue is empty!!!\n";

break;

case 4:

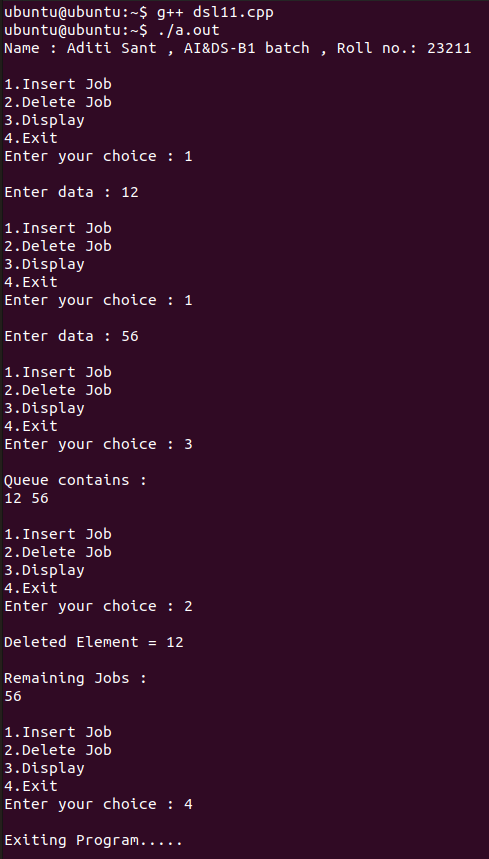
cout << "\nExiting Program.....\n\n";

}

} while (ch != 4);

return 0;

}

**OUTPUT**