# #11

**SOURCE CODE:**

#include<iostream> using namespace std; const int MAX = 10; struct job{

int data[MAX]; int front, rear;

};

class Queue{ struct job j; public: Queue(){

j.front = j.rear = -1;

}

bool isEmpty(); bool isFull();

void insertElement(); void removeElement(); void displayData();

};

bool Queue::isEmpty(){

if ((j.front == j.rear) || j.rear == -1){ return true;

}

return false;

}

bool Queue::isFull(){

return (j.rear == (MAX - 1)? true : false);

}

void Queue::insertElement(){ if (isFull()){

cout << "Queue Overflow" << endl;

}

else{

int x;

cout << "Enter Data : "; cin >> x; j.data[++j.rear] = x;

}

}

void Queue::removeElement(){ if (isEmpty()){

cout << "Queue Underflow" << endl;

}

else{

cout << "Element removed : " << j.data[j.front + 1] << endl; j.front++;

}

}

void Queue::displayData(){

for (int i = j.front + 1; i <= j.rear ; i++){ cout << j.data[i] << " " ;

}

cout << endl;

}

int main(){ Queue obj;

bool flag = true; while(flag){

cout << "CHOICES : " << endl;

cout << "1. Enter Data\n2. Remove Element\n3. Display Data\n4. Exit" << endl; int ch;

cout << "Enter your choice : "; cin >> ch;

switch (ch){ case 1:

obj.insertElement(); break;

case 2:

obj.removeElement(); break;

case 3:

obj.displayData(); break;

case 4:

cout << "Program ended !!" << endl; flag = false;

break; default:

cout << "Invalid choice "<< endl; break;

}

}

}