

```

#include <iostream>
#define MAXSIZE 25

using namespace std;

class Queue {
    int front,rear, q[MAXSIZE];

public:
    Queue(){
        front=rear=-1;
    }

    void insert();
    void del();
    void display();
    void isFull();
    void isEmpty();
};

void Queue::isFull() {
    if (rear == MAXSIZE - 1) {
        cout << "\nQUEUE IS FULL." << endl;
    } else {
        cout << "\nQUEUE IS NOT FULL." << endl;
    }
}

void Queue::isEmpty() {
    if (front == -1 && rear==-1) {
        cout << "\nQUEUE IS EMPTY." << endl;
    } else {
        cout << "\nQUEUE IS NOT EMPTY." << endl;
    }
}

void Queue::insert() {
    if (rear == MAXSIZE - 1) {
        cout << "\nQUEUE IS FULL." << endl;
    } else {

```

```

    int data;
    cout << "\nENTER THE DATA::>";
    cin >> data;
    rear++;
    q[rear] = data;
    if (front == -1) {
        front = 0;
    }
}
}

```

```

void Queue::del() {
    if (front == -1 && rear == -1) {
        cout << "\nQUEUE IS EMPTY." << endl;
    } else if (front == rear)
    {
        cout << "\n" << q[front] << " DATA IS DELETED." << endl;
        front=rear=-1;
    } else
    {
        cout << "\n" << q[front] << " DATA IS DELETED." << endl;
        front++;
    }
}
}

```

```

void Queue::display() {
    if (front == -1 && rear == -1) {
        cout << "\nQUEUE IS EMPTY." << endl;
    } else {
        cout << "\nQueue Elements: ";
        for (int i = front; i <= rear; i++) {
            cout << q[i] << "\t";
        }
        cout << endl;
    }
}
}

```

```

int main() {
    Queue queue;
    int choice;

```

```

    cout << "\n\nPROGRAM FOR IMPLEMENTATION OF QUEUE USING ARRAY (C++)\n";

```

```

while (true)
{
    cout << "\n1. INSERT \n2. DELETE \n3. DISPLAY \n4. CHECK IF FULL \n5. CHECK IF
EMPTY \n6. EXIT";
    cout << "\nENTER YOUR CHOICE::>";
    cin >> choice;

    switch (choice) {
        case 1:
            queue.insert();
            break;
        case 2:
            queue.del();
            break;
        case 3:
            queue.display();
            break;
        case 4:
            queue.isFull();
            break;
        case 5:
            queue.isEmpty();
            break;
        case 6:
            return 0;
        default:
            cout<<"wrong choice";

    }
}

return 0;
}

```