Q1.public class Stack {

private int maxSize;

private int top;

private int[] stackArray;

public Stack(int maxSize) {

this.maxSize = maxSize;

this.top = -1;

this.stackArray = new int[maxSize];

}

public void push(int item) {

if (top == maxSize - 1) {

System.out.println("Stack is full. Cannot push element.");

return;

}

stackArray[++top] = item;

}

public int pop() {

if (isEmpty()) {

System.out.println("Stack is empty. Cannot pop element.");

return -1; // or throw an exception

}

return stackArray[top--];

}

public boolean isEmpty() {

return top == -1;

}

}

Q2.public class Queue {

private Node front;

private Node rear;

public Queue() {

this.front = null;

this.rear = null;

}

public void enqueue(int item) {

Node newNode = new Node(item);

if (isEmpty()) {

front = newNode;

rear = newNode;

} else {

rear.next = newNode;

rear = newNode;

}

}

public int dequeue() {

if (isEmpty()) {

System.out.println("Queue is empty. Cannot dequeue element.");

return -1; // or throw an exception

}

int item = front.data;

front = front.next;

if (front == null) {

rear = null;

}

return item;

}

public boolean isEmpty() {

return front == null;

}

private static class Node {

private int data;

private Node next;

public Node(int data) {

this.data = data;

this.next = null;

}

}

}