Queue:

**Source Code:**

public class Queue<E> {  
  
 private Node<E> head;  
 private Node<E> tail;  
 private int size = 0;  
  
 private static class Node<E>{  
 E element;  
 Node<E> next;  
  
 public Node(E element){  
 this.element = element;  
 this.next = null;  
 }  
 }  
  
 public void insert(E element){  
 Node<E> newNode = new Node<>(element);  
 if (isEmpty()) {  
 head = newNode;  
 } else {  
 tail.next = newNode;  
 }  
 tail = newNode;  
 size++;  
 }  
  
 public E remove(){  
 if (isEmpty()) {  
 System.*out*.println("Query is empty");  
 return null;  
 }  
 E element = head.element;  
 head = head.next;  
  
 if (isEmpty()) {  
 tail = null;  
 }  
 size--;  
 return element;  
 }  
  
 public E peek(){  
 if (isEmpty()) {  
 System.*out*.println("Query is empty");  
 return null;  
 } else {  
 return head.element;  
 }  
 }  
  
 public boolean isEmpty(){  
 return size == 0;  
 }  
  
 public int size(){  
 return size;  
 }  
  
}

01)

**Source Code:**

import java.util.Scanner;  
  
public class StringParser {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter string: ");  
 String string = scanner.nextLine();  
 *stringParser*(string);  
 }  
  
 private static void stringParser(String string){  
 Queue<Character> newQueue = new Queue<>();  
 for (int i = 0; i < string.length(); i++) {  
 newQueue.insert(string.charAt(i));  
 }  
 for (int i = 0; i < string.length(); i++) {  
 System.*out*.print(newQueue.remove());  
  
 }  
 }  
}

**Output:**

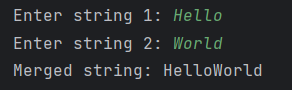


02)

**Source Code:**

import java.util.Scanner;  
  
public class StringMerger {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter string 1: ");  
 String str1 = scanner.next();  
 System.*out*.print("Enter string 2: ");  
 String str2 = scanner.next();  
 String mergerdString = *stringMerger*(str1, str2);  
 System.*out*.println("Merged string: " + mergerdString);  
 }  
  
 public static String stringMerger(String str1, String str2){  
 Queue<String> newQueue =new Queue<>();  
 newQueue.insert(str1);  
 newQueue.insert(str2);  
 return (newQueue.remove() + newQueue.remove());  
 }  
}

**Output:**



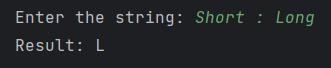
**03)**

**Source Code:**

import java.util.Scanner;  
  
public class StringComparison {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter the string: ");  
 String string = scanner.nextLine();  
 Character output = *stringComparison*(string);  
 System.*out*.println("Result: " + output);  
 }  
  
 private static Character stringComparison(String string){  
 Queue<String> newQueue = new Queue<>();  
 if (string.contains(":")) {  
 String[] arrayOfStr = string.split(" : ");  
 for (String str : arrayOfStr){  
 newQueue.insert(str);  
 }  
 String string1 = newQueue.remove();  
 String string2 = newQueue.remove();  
  
 if (string1.length() > string2.length()) {  
 return 'L';  
 } else if (string1.length() < string1.length()) {  
 return 'R';  
 } else if (string1.length() == string1.length() && string1.equals(string2)) {  
 return 'S';  
 } else if (string1.length() == string1.length()) {  
 return 'D';  
 } else {  
 return null;  
 }  
 } else {  
 return 'N';  
 }  
 }  
}

**Output:**









04)

**Source Code:**

import java.util.Scanner;  
  
public class NumberChanger {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter number: ");  
 String number = scanner.next();  
 String changedNumber = *numberChanger*(number);  
 System.*out*.println(changedNumber);  
 }  
  
 private static String numberChanger(String number){  
 Queue<String> newQueue = new Queue<>();  
 int numberLen = number.length();  
 int mid = numberLen/2;  
 String[] arrayOfNumbers = new String[]{number.substring(mid), number.substring(0, mid)};  
 for (String num : arrayOfNumbers){  
 newQueue.insert(num);  
 }  
 return (newQueue.remove() + newQueue.remove());  
 }  
}

**Output:**



05)

**Source Code:**

import java.util.Scanner;  
  
public class VowelFinder {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter a string: ");  
 String string = scanner.nextLine();  
 System.*out*.print("Vowels: ");  
 *vowelFinder*(string);  
 }  
  
 private static void vowelFinder(String string){  
 Queue<Character> newQueue = new Queue<>();  
 for (int i = 0; i < string.length(); i++) {  
 char letter = string.charAt(i);  
 switch (letter) {  
 case 'A', 'U', 'O', 'I', 'E', 'a', 'e', 'i', 'o', 'u' -> newQueue.insert(letter);  
 default -> {  
 continue;  
 }  
 }  
 }  
 while (!newQueue.isEmpty()){  
 System.*out*.print(newQueue.remove() + " ");  
 }  
 }  
}

**Output:**



06)

**Source Code:**

import java.util.Scanner;  
  
public class NumberInterleave {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter number sequence: ");  
 String numberSequence = scanner.nextLine();  
 String changedNumber = *numberInterleave*(numberSequence);  
 System.*out*.println(changedNumber);  
 }  
  
 private static String numberInterleave(String number){  
 Queue<String> newQueue1 = new Queue<>();  
 Queue<String> newQueue2 = new Queue<>();  
 int count = 1;  
 String[] arrayOfNumbers = number.split(" ");  
 int numOfElements = arrayOfNumbers.length;  
 int mid = numOfElements / 2;  
 for (String num : arrayOfNumbers){  
 if (count <= mid){  
 newQueue1.insert(num);  
 count++;  
 } else if (count <= arrayOfNumbers.length){  
 newQueue2.insert(num);  
 count++;  
 }  
 }  
 String interleavedNumber = "";  
 for (int i = 0; i < mid; i++) {  
 interleavedNumber += newQueue1.remove() + " " + newQueue2.remove() + " ";  
 }  
 return interleavedNumber;  
 }  
}

**Output:**



07)

**Source Code:**

import java.util.Scanner;  
  
class Student{  
 private int stdNo;  
 private int mathsMarks;  
 private int chemistryMarks;  
 private int physicsMarks;  
 private float totalMarks;  
 private float avgMarks;  
  
 public Student(int stdNo, int mathsMarks, int chemistryMarks, int physicsMarks) {  
 this.stdNo = stdNo;  
 this.mathsMarks = mathsMarks;  
 this.chemistryMarks = chemistryMarks;  
 this.physicsMarks = physicsMarks;  
 this.totalMarks = mathsMarks + chemistryMarks + physicsMarks;  
 this.avgMarks = totalMarks / 3;  
 }  
  
 public int getStdNo() {  
 return stdNo;  
 }  
  
 public void setStdNo(int stdNo) {  
 this.stdNo = stdNo;  
 }  
  
 public int getMathsMarks() {  
 return mathsMarks;  
 }  
  
 public void setMathsMarks(int mathsMarks) {  
 this.mathsMarks = mathsMarks;  
 }  
  
 public int getChemistryMarks() {  
 return chemistryMarks;  
 }  
  
 public void setChemistryMarks(int chemistryMarks) {  
 this.chemistryMarks = chemistryMarks;  
 }  
  
 public int getPhysicsMarks() {  
 return physicsMarks;  
 }  
  
 public void setPhysicsMarks(int physicsMarks) {  
 this.physicsMarks = physicsMarks;  
 }  
  
 public float getTotalMarks() {  
 return totalMarks;  
 }  
  
 public void setTotalMarks(float totalMarks) {  
 this.totalMarks = totalMarks;  
 }  
  
 public float getAvgMarks() {  
 return avgMarks;  
 }  
  
 public void setAvgMarks(float avgMarks) {  
 this.avgMarks = avgMarks;  
 }  
}  
  
public class MarksCalculator {  
 public static void main(String[] args) {  
 Queue<Student> studentQueue = new Queue<>();  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("How much student do you want to add to database: ");  
 int noOfStudents = scanner.nextInt();  
 System.*out*.println();  
  
 for (int i = 0; i < noOfStudents; i++) {  
 System.*out*.print("Enter the Student No: ");  
 int stdNo = scanner.nextInt();  
 System.*out*.print("Enter the Maths Marks: ");  
 int mathsMarks = scanner.nextInt();  
 System.*out*.print("Enter the Chemistry Marks: ");  
 int chemistryMarks = scanner.nextInt();  
 System.*out*.print("Enter the Physics Marks: ");  
 int physicsMarks = scanner.nextInt();  
 System.*out*.println();  
  
 studentQueue.insert(new Student(stdNo, mathsMarks, chemistryMarks, physicsMarks));  
 }  
  
 while (!studentQueue.isEmpty()){  
 Student student = studentQueue.remove();  
 System.*out*.println("Average mark of student number " + student.getStdNo() + " is " + student.getAvgMarks());  
 }  
 }  
  
  
  
}

**Output:**

