import java.util.Random; // Importa la clase Random para generar números aleatorios

public class App {

    public static void main(String[] args) throws Exception {

        // Crea un arreglo de 5 elementos llamado arrregoUno

        int[] arrregoUno = new int[5];

        // Llena el arreglo con números aleatorios entre 1 y 10

        for (int i = 0; i < arrregoUno.length; i++) {

            Random random = new Random(); // Crea una instancia de Random

            int x = random.nextInt(10) + 1; // Genera un número aleatorio entre 1 y 10

            arrregoUno[i] = x; // Asigna el número aleatorio al arreglo

        }

        // Imprime el mensaje del tipo de arreglo

        System.out.print("El arreglo uno es: ");

        for (int i = 0; i < arrregoUno.length; i++) {

            System.out.print(arrregoUno[i] + " "); // Imprime cada elemento del arreglo

        }

        System.out.println(" "); // Salto de línea

        // Crea otro arreglo de 5 elementos llamado arregloDos

        int[] arregloDos = new int[5];

        // Llena el segundo arreglo con números aleatorios entre 1 y 10

        for (int j = 0; j < arregloDos.length; j++) {

            Random random = new Random(); // Crea una nueva instancia de Random

            int x = random.nextInt(10) + 1; // Genera un número aleatorio entre 1 y 10

            arregloDos[j] = x; // Asigna el número aleatorio al arreglo

        }

        // Imprime el contenido del segundo arreglo

        System.out.print("El arreglo dos es: ");

        for (int i = 0; i < arregloDos.length; i++) {//For para recorrer los elementos

            System.out.print(arregloDos[i] + " "); // Imprime cada elemento del arreglo

        }

        System.out.println(""); // Salto de línea

        // Imprime un arreglo con elementos repetidos

        System.out.print("El con repetido: ");

        int[] arregloConRepetidos = { 1, 2, 2, 5, 6, 1, 3, 3 }; // Arreglo con valores duplicados

        for (int i = 0; i < arregloConRepetidos.length; i++) {//For para recorrer los elementos

            System.out.print(arregloConRepetidos[i] + " "); // Imprime cada elemento del arreglo

        }

        System.out.println(""); // Salto de línea

        // Crea una instancia de la clase Operaciones

        Operaciones op = new Operaciones();

        // Calcula y muestra la suma de los dos arreglos

        System.out.print("La suma es: ");

        int suma[] = op.suma(arrregoUno, arregloDos); // Llama al método suma

        for (int i = 0; i < suma.length; i++) {//For para recorrer los elementos

            System.out.print(+suma[i] + " "); // Imprime cada elemento del arreglo de suma

        }

        System.out.println(" "); // Salto de línea

        // Calcula y muestra la multiplicación de los dos arreglos

        System.out.print("La Multiplicación es: ");

        int multiplicacion[] = op.multiplicacion(arrregoUno, arregloDos); // Llama al método multiplicacion

        for (int k = 0; k < multiplicacion.length; k++) {//For para recorrer los elementos

            System.out.print(+multiplicacion[k] + " "); // Imprime cada elemento del arreglo de multiplicación

        }

        System.out.println(" "); // Salto de línea

        // Calcula y muestra la concatenación de los dos arreglos

        System.out.print("La concatenacion es: ");

        int concadenacion[] = op.concadenacion(arrregoUno, arregloDos); // Llama al método concatenacion

        for (int k = 0; k < concadenacion.length; k++) {//For para recorrer los elementos

            System.out.print(+concadenacion[k] + " "); // Imprime cada elemento del arreglo concatenado

        }

        // Elimina los valores repetidos del arreglo y muestra el resultado

        int[] arregloSinRepetidos = op.eliminarRepetidos(arregloConRepetidos); // Llama al método eliminarRepetidos

        System.out.println(""); // Salto de línea

        System.out.print("Arreglo sin repetidos: ");

        for (int i = 0; i < arregloSinRepetidos.length; i++) {//For para recorrer los elementos

            if (i > 0) {//condicional para evaluar en la posicion i

                System.out.print(", "); // Imprime una coma entre elementos

            }

            System.out.print(arregloSinRepetidos[i]); // Imprime cada elemento del arreglo sin repetidos

        }

        System.out.println(); // Salto de línea

    }

}

public class Operaciones {

    // Método para sumar dos arreglos

    public int[] suma(int arrregoUno[], int arregloDos[]) {

        // Crea un arreglo para almacenar el resultado de la suma

        int resultadoSuma[] = new int[5];

        // Recorre cada elemento de los arreglos

        for (int i = 0; i < resultadoSuma.length; i++) {

            resultadoSuma[i] = arrregoUno[i] + arregloDos[i]; // Suma los elementos correspondientes

        }

        return resultadoSuma; // Retorna el arreglo con el resultado de la suma

    }

    // Método para multiplicar dos arreglos

    public int[] multiplicacion(int arrregoUno[], int arregloDos[]) {

        // Crea un arreglo para almacenar el resultado de la multiplicación

        int resultadoMultiplicacion[] = new int[5];

        // Recorre cada elemento de los arreglos

        for (int i = 0; i < resultadoMultiplicacion.length; i++) {

            resultadoMultiplicacion[i] = arrregoUno[i] \* arregloDos[i]; // Multiplica los elementos correspondientes

        }

        return resultadoMultiplicacion; // Retorna el arreglo con el resultado de la multiplicación

    }

    // Método para concatenar dos arreglos

    public int[] concadenacion(int arregloUno[], int arregloDos[]) {

        // Crea un arreglo para almacenar el resultado de la concatenación

        int resultadoConcadenacion[] = new int[arregloDos.length + arregloUno.length];

        // Copia los elementos del primer arreglo

        for (int i = 0; i < arregloUno.length; i++) {//For para recorrer elementos

            resultadoConcadenacion[i] = arregloUno[i]; // Asigna el elemento del primer arreglo

        }

        // Copia los elementos del segundo arreglo

        for (int j = 0; j < arregloDos.length; j++) {//For para recorrer elementos

            resultadoConcadenacion[j + arregloUno.length] = arregloDos[j]; // Asigna el elemento del segundo arreglo

        }

        return resultadoConcadenacion; // Retorna el arreglo con la concatenación de ambos

    }

    // Método para eliminar elementos repetidos de un arreglo

    public int[] eliminarRepetidos(int[] arr) {

        // Crea un arreglo temporal del mismo tamaño que el original

        int[] temporal = new int[arr.length];

        // Variable contador para llevar el número de elementos únicos encontrados

        int contador = 0;

        for (int i = 0; i < arr.length; i++) {//For para recorrer elementos

            // Variable para verificar si el valor actual ya fue encontrado

            boolean repetido = false;//declaro repetido

            // Bucle interno para comparar el elemento actual de `arr` con los elementos ya añadidos al arreglo temporal

            for (int j = 0; j < contador; j++) {//For para recorrer elementos

                // Si el valor actual de `arr[i]` ya está en `temporal`, lo marcamos como repetido

                if (arr[i] == temporal[j]) {//evalua que en la posicion sean iguales

                    repetido = true;// se cambia le valor de boleano a true

                    break; // Salimos del bucle si encontramos un repetido

                }

            }

            // Si no se encontró repetido, lo añadimos al arreglo temporal

            if (!repetido) {//condicional para verificar que resultado es diferente

                temporal[contador] = arr[i]; // Agrega el valor actual de `arr[i]` al arreglo temporal

                contador++; // Incrementa el contador de elementos únicos

            }

        }

        // Crea un nuevo arreglo con el tamaño exacto del número de elementos únicos

        int[] resultado = new int[contador];

        // Copia los valores únicos del arreglo temporal al nuevo arreglo resultado

        for (int i = 0; i < contador; i++) {//For para recorrer los elementos

            resultado[i] = temporal[i];//Tepral en la posicion i lo guarda en resultado

        }

        // Retorna el nuevo arreglo que contiene solo los valores únicos

        return resultado;

    }

}