

Caso 1: 25 puntos

Pide un número de cursos.

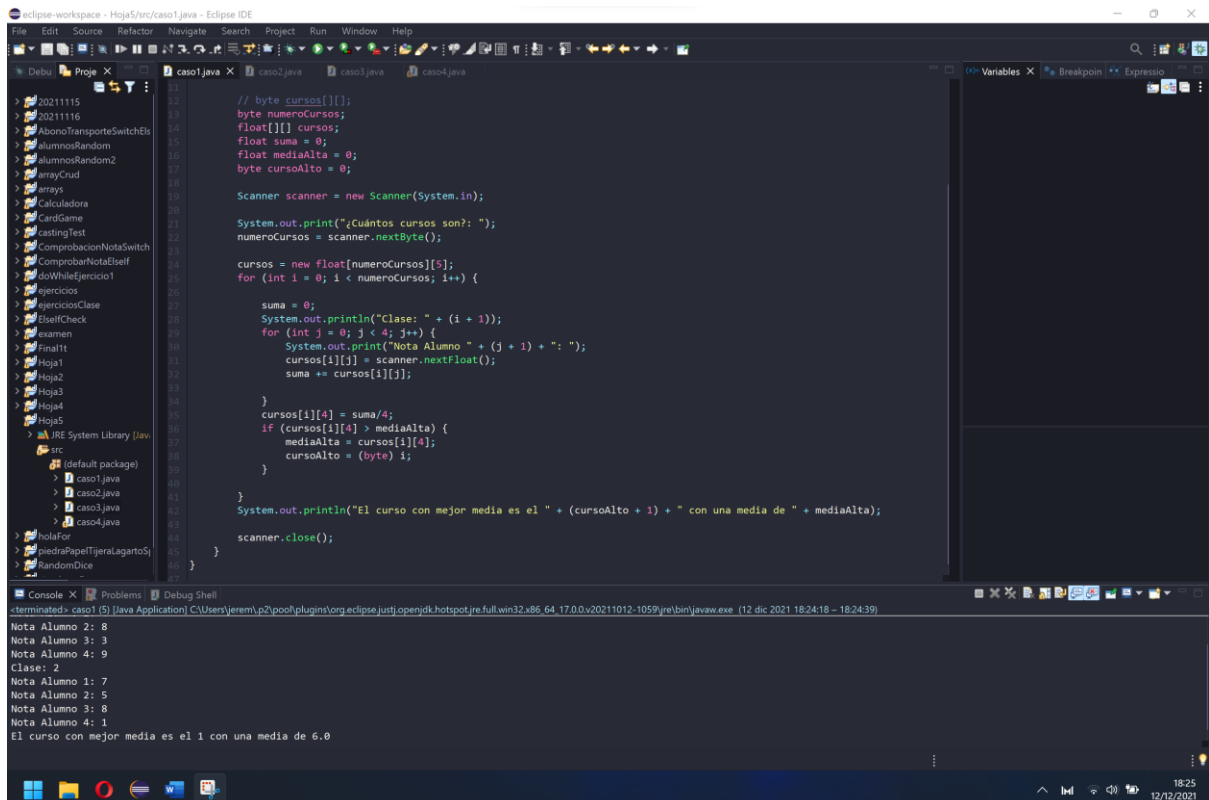
En cada curso hay 4 alumnos.

Pide las notas de cada alumno.

Calcula la media.

Muestra el curso con mejor media.

PRUEBAS:



```

1 // byte cursos[][];
2 byte numeroCursos;
3 float[][] cursos;
4 float suma = 0;
5 float mediaAlta = 0;
6 byte cursoAlto = 0;
7
8 Scanner scanner = new Scanner(System.in);
9
10 System.out.print("¿Cuántos cursos son?: ");
11 numeroCursos = scanner.nextByte();
12
13 cursos = new float[numeroCursos][5];
14 for (int i = 0; i < numeroCursos; i++) {
15     suma = 0;
16     System.out.println("Clase: " + (i + 1));
17     for (int j = 0; j < 4; j++) {
18         System.out.print("Nota Alumno " + (j + 1) + ": ");
19         cursos[i][j] = scanner.nextFloat();
20         suma += cursos[i][j];
21     }
22     cursos[i][4] = suma/4;
23     if (cursos[i][4] > mediaAlta) {
24         mediaAlta = cursos[i][4];
25         cursoAlto = (byte) i;
26     }
27 }
28 System.out.println("El curso con mejor media es el " + (cursoAlto + 1) + " con una media de " + mediaAlta);
29 scanner.close();
30 }

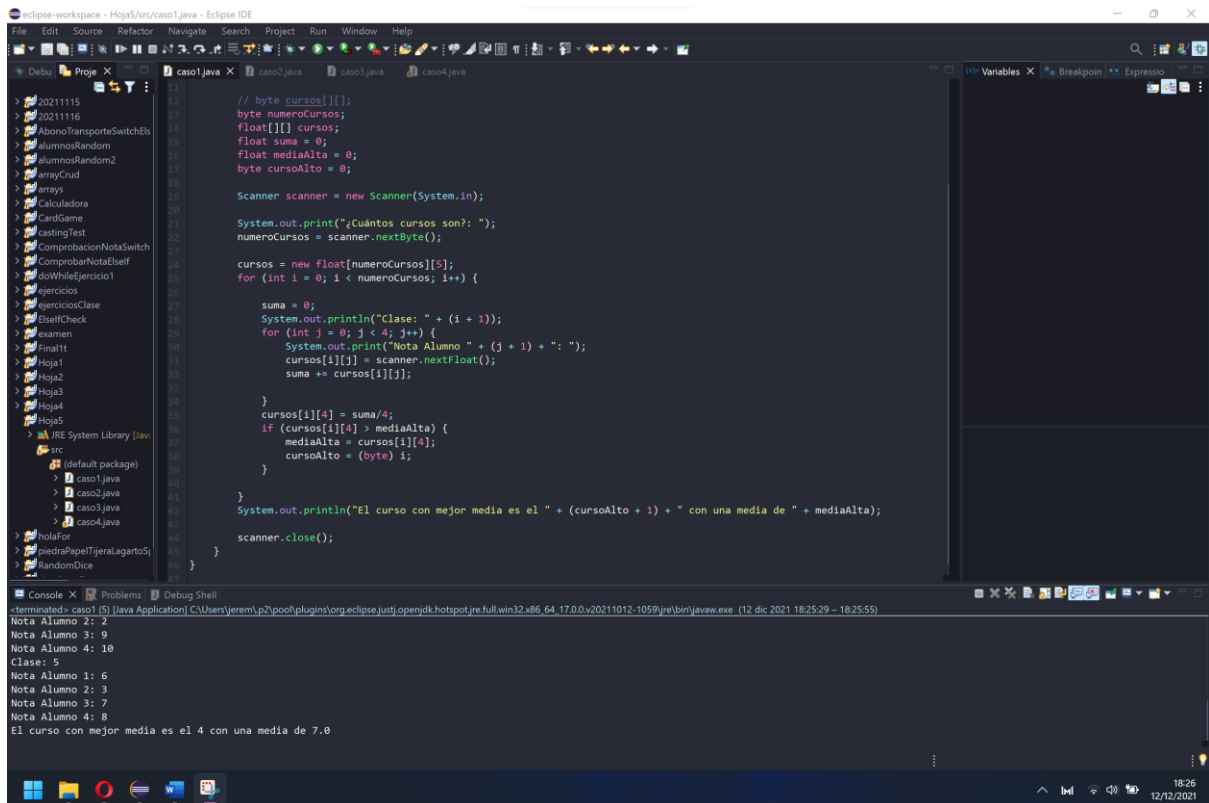
```

Console Output:

```

<terminated> caso1 (5) [Java Application] C:\Users\jerem\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.0.v20211012-1059\jre\bin\javaw.exe (12 dic 2021 18:24:18 - 18:24:39)
Nota Alumno 2: 8
Nota Alumno 3: 3
Nota Alumno 4: 9
Clase: 2
Nota Alumno 1: 7
Nota Alumno 2: 5
Nota Alumno 3: 8
Nota Alumno 4: 1
El curso con mejor media es el 1 con una media de 6.0

```



```

11 // byte cursos[][];
12 byte numeroCursos;
13 float[][] cursos;
14 float suma = 0;
15 float mediaAlta = 0;
16 byte cursoAlto = 0;
17
18 Scanner scanner = new Scanner(System.in);
19
20 System.out.print("¿Cuántos cursos son?: ");
21 numeroCursos = scanner.nextInt();
22
23 cursos = new float[numeroCursos][5];
24 for (int i = 0; i < numeroCursos; i++) {
25     suma = 0;
26     System.out.println("Clase: " + (i + 1));
27     for (int j = 0; j < 4; j++) {
28         System.out.print("Nota Alumno " + (j + 1) + ": ");
29         cursos[i][j] = scanner.nextFloat();
30         suma += cursos[i][j];
31     }
32     cursos[i][4] = suma/4;
33     if (cursos[i][4] > mediaAlta) {
34         mediaAlta = cursos[i][4];
35         cursoAlto = (byte) i;
36     }
37 }
38 System.out.println("El curso con mejor media es el " + (cursoAlto + 1) + " con una media de " + mediaAlta);
39 scanner.close();
40
41 }
42
43
44
45
46
47

```

Console X | Problems | Debug Shell

```

terminated - caso1 [5] (Java Application) C:\Users\jerem.p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.0.v20211012-1059\jre\bin\javaw.exe (12 dic 2021 18:25:29 - 18:25:55)
Nota Alumno 2: 2
Nota Alumno 3: 9
Nota Alumno 4: 10
Clase: 5
Nota Alumno 1: 6
Nota Alumno 2: 3
Nota Alumno 3: 7
Nota Alumno 4: 8
El curso con mejor media es el 4 con una media de 7.0

```

Caso 2: 25 punto

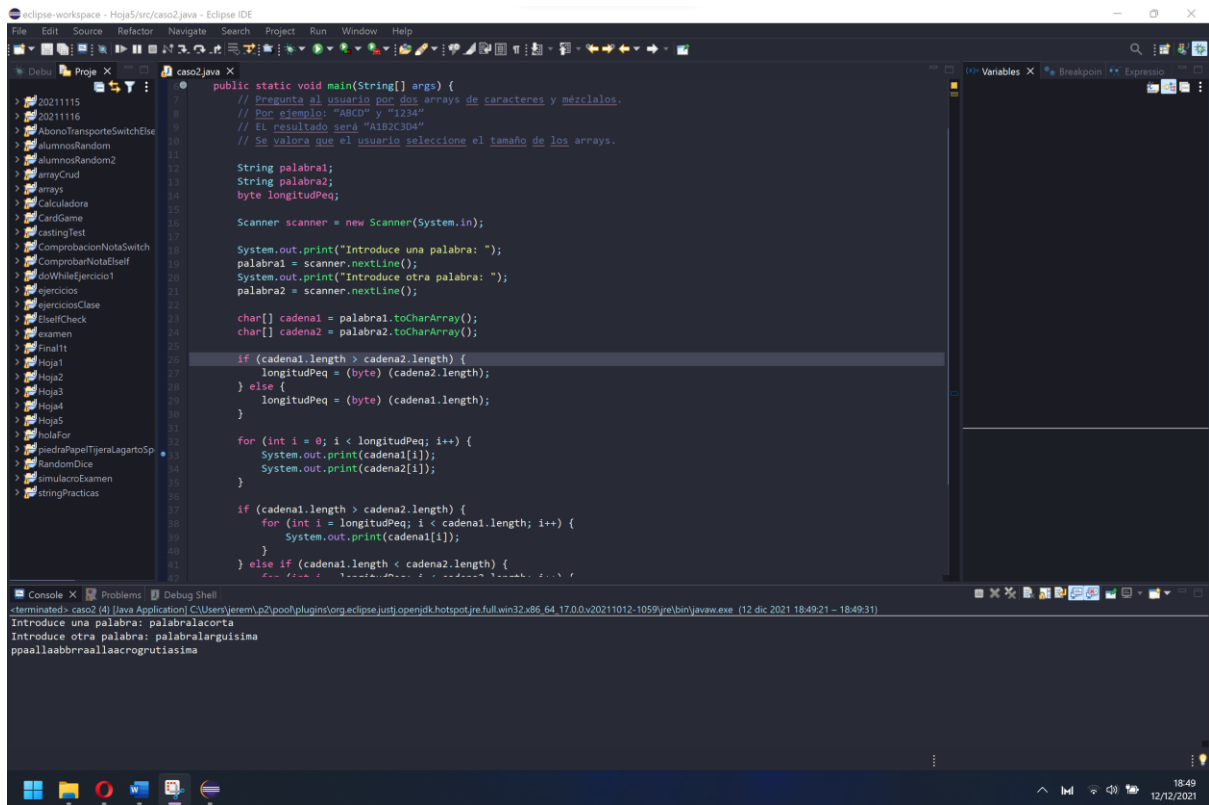
Pregunta al usuario por dos arrays de caracteres y mézclalos.

Por ejemplo: "ABCD" y "1234"

EL resultado será "A1B2C3D4"

Se valora que el usuario seleccione el tamaño de los arrays.

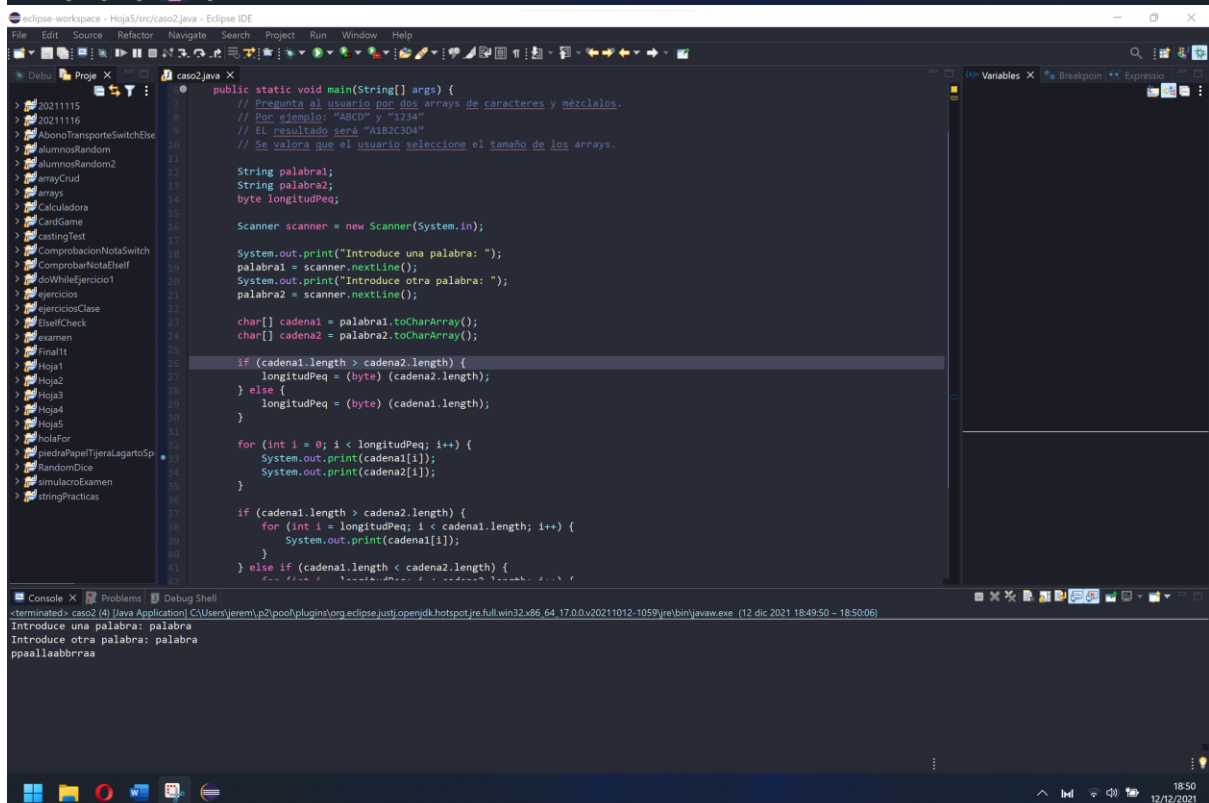
PRUEBAS:



```
public static void main(String[] args) {  
    // Pregunta al usuario por dos arrays de caracteres y mércalos.  
    // Por ejemplo: "ABCD" y "1234"  
    // El resultado será "A1B2C3D4"  
    // Se valora que el usuario seleccione el tamaño de los arrays.  
  
    String palabra1;  
    String palabra2;  
    byte longitudPeq;  
  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.print("Introduce una palabra: ");  
    palabra1 = scanner.nextLine();  
    System.out.print("Introduce otra palabra: ");  
    palabra2 = scanner.nextLine();  
  
    char[] cadena1 = palabra1.toCharArray();  
    char[] cadena2 = palabra2.toCharArray();  
  
    if (cadena1.length > cadena2.length) {  
        longitudPeq = (byte) (cadena2.length);  
    } else {  
        longitudPeq = (byte) (cadena1.length);  
    }  
  
    for (int i = 0; i < longitudPeq; i++) {  
        System.out.print(cadena1[i]);  
        System.out.print(cadena2[i]);  
    }  
  
    if (cadena1.length > cadena2.length) {  
        for (int i = longitudPeq; i < cadena1.length; i++) {  
            System.out.print(cadena1[i]);  
        }  
    } else if (cadena1.length < cadena2.length) {  
        for (int i = longitudPeq; i < cadena2.length; i++) {  
            System.out.print(cadena2[i]);  
        }  
    }  
}
```

Console Output:

```
<terminated> caso2 (4) [Java Application] C:\Users\jerem\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full\win32\x86_64\17.0.0.v20211012-1059\jre\bin\java.exe (12 dic 2021 18:49:21 - 18:49:31)  
Introduce una palabra: ppaallaabrrraa  
Introduce otra palabra: laacrogrutiasima  
ppaallaabrrraa laacrogrutiasima
```



```
public static void main(String[] args) {  
    // Pregunta al usuario por dos arrays de caracteres y mércalos.  
    // Por ejemplo: "ABCD" y "1234"  
    // El resultado será "A1B2C3D4"  
    // Se valora que el usuario seleccione el tamaño de los arrays.  
  
    String palabra1;  
    String palabra2;  
    byte longitudPeq;  
  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.print("Introduce una palabra: ");  
    palabra1 = scanner.nextLine();  
    System.out.print("Introduce otra palabra: ");  
    palabra2 = scanner.nextLine();  
  
    char[] cadena1 = palabra1.toCharArray();  
    char[] cadena2 = palabra2.toCharArray();  
  
    if (cadena1.length > cadena2.length) {  
        longitudPeq = (byte) (cadena2.length);  
    } else {  
        longitudPeq = (byte) (cadena1.length);  
    }  
  
    for (int i = 0; i < longitudPeq; i++) {  
        System.out.print(cadena1[i]);  
        System.out.print(cadena2[i]);  
    }  
  
    if (cadena1.length > cadena2.length) {  
        for (int i = longitudPeq; i < cadena1.length; i++) {  
            System.out.print(cadena1[i]);  
        }  
    } else if (cadena1.length < cadena2.length) {  
        for (int i = longitudPeq; i < cadena2.length; i++) {  
            System.out.print(cadena2[i]);  
        }  
    }  
}
```

Console Output:

```
<terminated> caso2 (4) [Java Application] C:\Users\jerem\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full\win32\x86_64\17.0.0.v20211012-1059\jre\bin\java.exe (12 dic 2021 18:49:50 - 18:50:06)  
Introduce una palabra: ppaallaabrrraa  
Introduce otra palabra: palabra  
ppaallaabrrraa
```

```
public static void main(String[] args) {  
    // Pregunta al usuario por dos arrays de caracteres y mércialos.  
    // Por ejemplo: "ABCD" y "1234"  
    // El resultado será "A1B2C3D4"  
    // Se valora que el usuario seleccione el tamaño de los arrays.  
  
    String palabra1;  
    String palabra2;  
    byte longitudPeg;  
  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.print("Introduce una palabra: ");  
    palabra1 = scanner.nextLine();  
    System.out.print("Introduce otra palabra: ");  
    palabra2 = scanner.nextLine();  
  
    char[] cadena1 = palabra1.toCharArray();  
    char[] cadena2 = palabra2.toCharArray();  
  
    if (cadena1.length > cadena2.length) {  
        longitudPeg = (byte) (cadena2.length);  
    } else {  
        longitudPeg = (byte) (cadena1.length);  
    }  
  
    for (int i = 0; i < longitudPeg; i++) {  
        System.out.print(cadena1[i]);  
        System.out.print(cadena2[i]);  
    }  
  
    if (cadena1.length > cadena2.length) {  
        for (int i = longitudPeg; i < cadena1.length; i++) {  
            System.out.print(cadena1[i]);  
        }  
    } else if (cadena1.length < cadena2.length) {  
        for (int i = longitudPeg; i < cadena2.length; i++) {  
            System.out.print(cadena2[i]);  
        }  
    }  
}
```

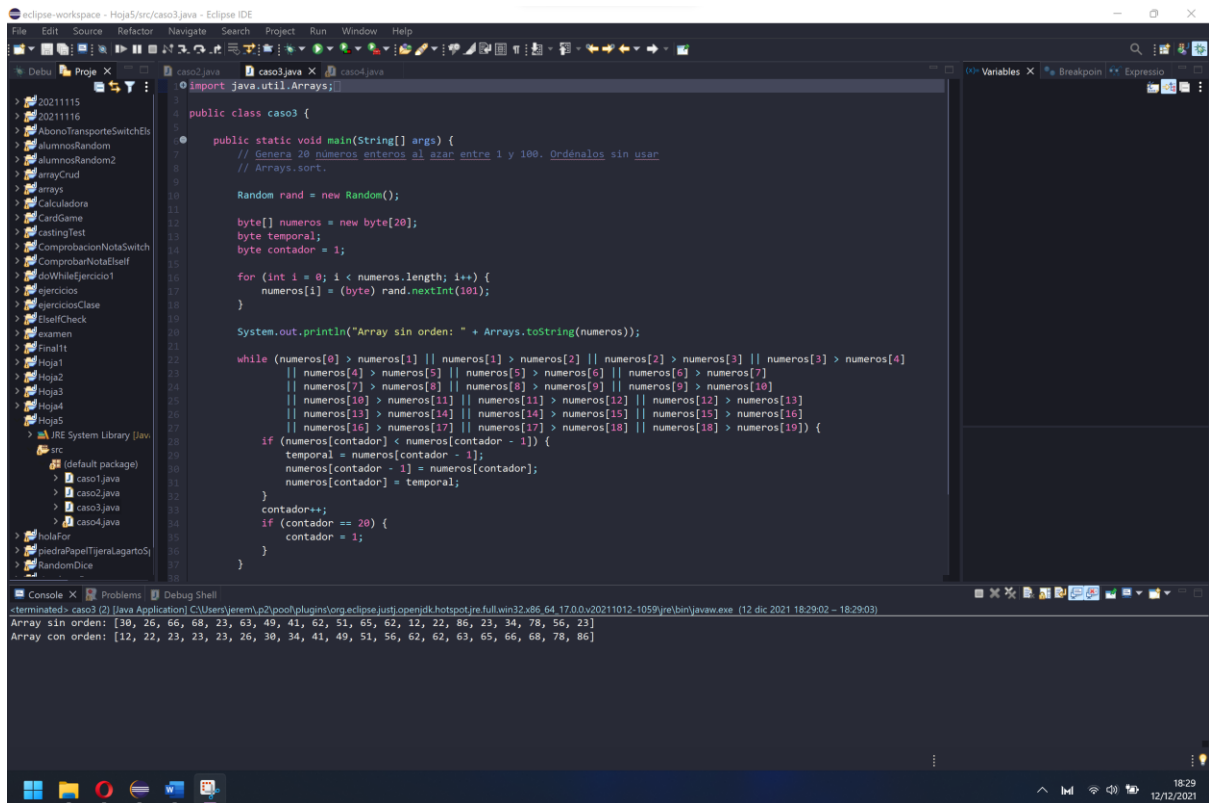
Console X Problems Debug Shell
terminated: caso2 (4) [Java Application] C:\Users\jerem.p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.0.v20211012-1059\jre\bin\java.exe (12 dic 2021 18:48:28 - 18:48:47)
Introduce una palabra: palabrarauguissima
Introduce otra palabra: palabracorta
ppaallaabrraalcaorrgtuaisima

Caso 3: 25 punto

Genera **20** números enteros al azar entre 1 y 100. Ordénalos sin usar Arrays.sort.

Ojo con buscar la solución en internet. Se pueden identificar con facilidad.

PRUEBAS:



```

import java.util.Arrays;

public class caso3 {

    public static void main(String[] args) {
        // Genera 20 números enteros al azar entre 1 y 100. Ordénalos sin usar
        // Arrays.sort.

        Random rand = new Random();

        byte[] numeros = new byte[20];
        byte temporal;
        byte contador = 1;

        for (int i = 0; i < numeros.length; i++) {
            numeros[i] = (byte) rand.nextInt(101);
        }

        System.out.println("Array sin orden: " + Arrays.toString(numeros));

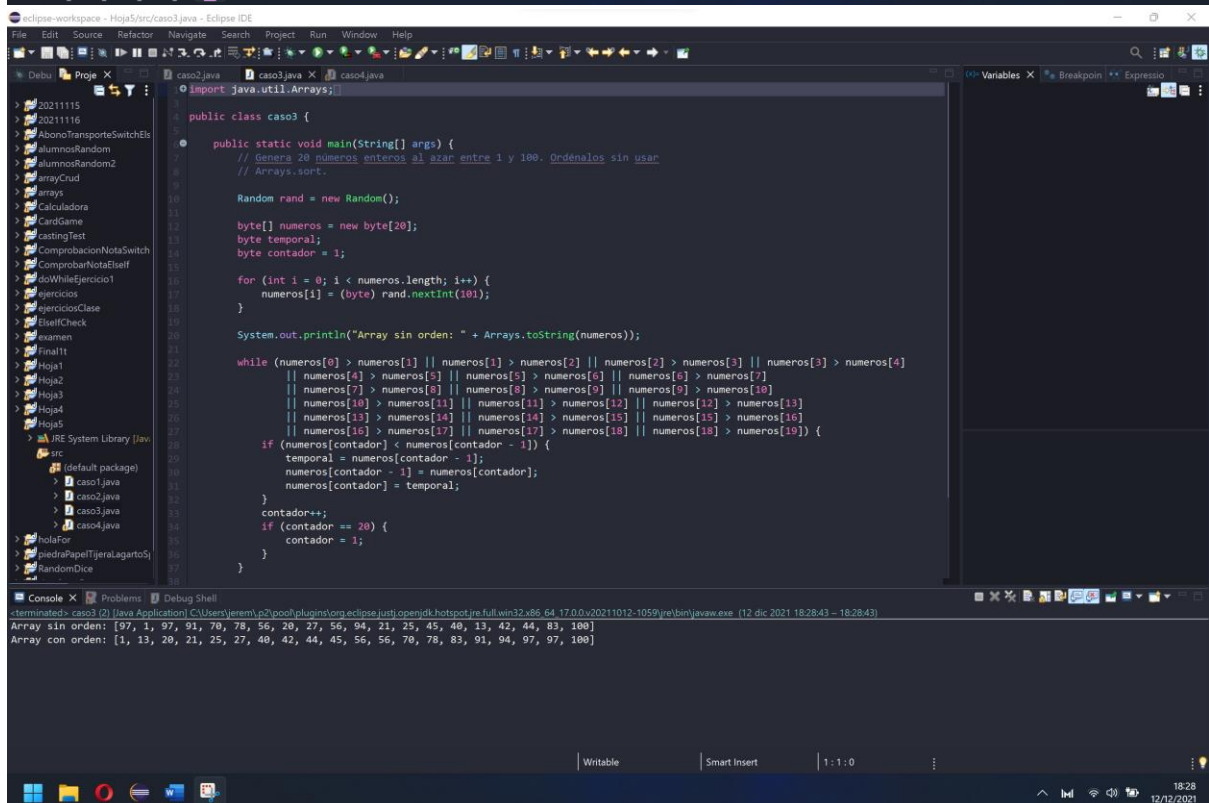
        while (numeros[0] > numeros[1] || numeros[1] > numeros[2] || numeros[2] > numeros[3] || numeros[3] > numeros[4]
            || numeros[4] > numeros[5] || numeros[5] > numeros[6] || numeros[6] > numeros[7]
            || numeros[7] > numeros[8] || numeros[8] > numeros[9] || numeros[9] > numeros[10]
            || numeros[10] > numeros[11] || numeros[11] > numeros[12] || numeros[12] > numeros[13]
            || numeros[13] > numeros[14] || numeros[14] > numeros[15] || numeros[15] > numeros[16]
            || numeros[16] > numeros[17] || numeros[17] > numeros[18] || numeros[18] > numeros[19]) {

            if (numeros[contador] < numeros[contador - 1]) {
                temporal = numeros[contador - 1];
                numeros[contador - 1] = numeros[contador];
                numeros[contador] = temporal;
            }
            contador++;
            if (contador == 20) {
                contador = 1;
            }
        }

        System.out.println("Array con orden: " + Arrays.toString(numeros));
    }
}

```

Array sin orden: [30, 26, 66, 68, 23, 63, 49, 41, 62, 51, 65, 62, 12, 22, 86, 23, 34, 78, 56, 23]
 Array con orden: [12, 22, 23, 23, 23, 26, 30, 34, 41, 49, 51, 56, 62, 62, 63, 65, 66, 68, 78, 86]



```

import java.util.Arrays;

public class caso3 {

    public static void main(String[] args) {
        // Genera 20 números enteros al azar entre 1 y 100. Ordénalos sin usar
        // Arrays.sort.

        Random rand = new Random();

        byte[] numeros = new byte[20];
        byte temporal;
        byte contador = 1;

        for (int i = 0; i < numeros.length; i++) {
            numeros[i] = (byte) rand.nextInt(101);
        }

        System.out.println("Array sin orden: " + Arrays.toString(numeros));

        while (numeros[0] > numeros[1] || numeros[1] > numeros[2] || numeros[2] > numeros[3] || numeros[3] > numeros[4]
            || numeros[4] > numeros[5] || numeros[5] > numeros[6] || numeros[6] > numeros[7]
            || numeros[7] > numeros[8] || numeros[8] > numeros[9] || numeros[9] > numeros[10]
            || numeros[10] > numeros[11] || numeros[11] > numeros[12] || numeros[12] > numeros[13]
            || numeros[13] > numeros[14] || numeros[14] > numeros[15] || numeros[15] > numeros[16]
            || numeros[16] > numeros[17] || numeros[17] > numeros[18] || numeros[18] > numeros[19]) {

            if (numeros[contador] < numeros[contador - 1]) {
                temporal = numeros[contador - 1];
                numeros[contador - 1] = numeros[contador];
                numeros[contador] = temporal;
            }
            contador++;
            if (contador == 20) {
                contador = 1;
            }
        }

        System.out.println("Array con orden: " + Arrays.toString(numeros));
    }
}

```

Array sin orden: [97, 1, 97, 70, 78, 56, 20, 27, 56, 94, 21, 25, 45, 40, 13, 42, 44, 83, 100]
 Array con orden: [1, 13, 20, 21, 25, 27, 40, 42, 44, 45, 56, 56, 70, 78, 83, 91, 94, 97, 100]

Caso 4: 25 puntos

Genera un boleto de bingo al azar siguiendo este patrón:

1	11	20	32	44	50	61	73	81	93
---	----	----	----	----	----	----	----	----	----

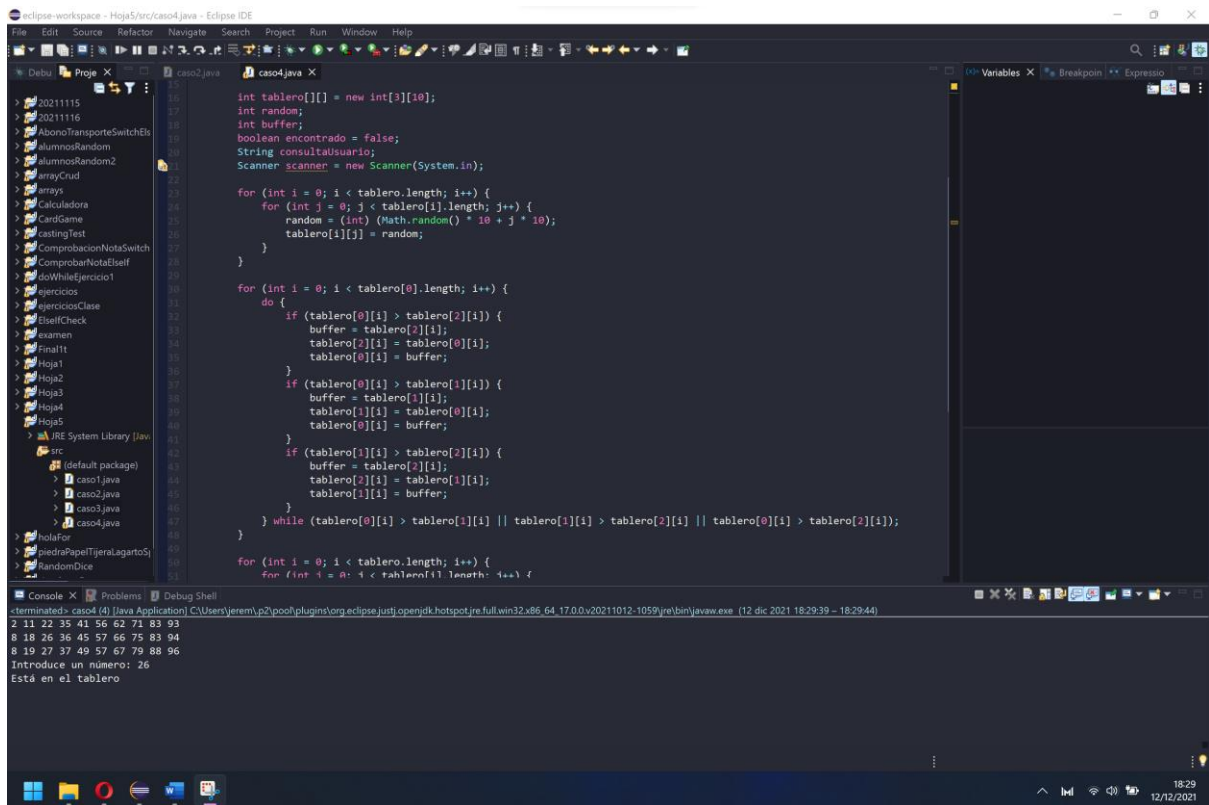
4	13	21	38	47	53	66	74	87	94
9	18	22	39	49	55	68	76	88	98

Se debe mostrar por consola en un formato que permita su lectura de manera razonable.

Se debe almacenar en un array bidimensional.

Pide un número al usuario y devuelve si se encuentra en su cartón.

PRUEBAS:



```

1  int tablero[][] = new int[3][10];
2  int random;
3  int buffer;
4  boolean encontrado = false;
5  String consultaUsuario;
6  Scanner scanner = new Scanner(System.in);
7
8  for (int i = 0; i < tablero.length; i++) {
9      for (int j = 0; j < tablero[i].length; j++) {
10         random = (int) (Math.random() * 10 + j * 10);
11         tablero[i][j] = random;
12     }
13 }
14
15 for (int i = 0; i < tablero[0].length; i++) {
16     do {
17         if (tablero[0][i] > tablero[2][i]) {
18             buffer = tablero[2][i];
19             tablero[2][i] = tablero[0][i];
20             tablero[0][i] = buffer;
21         }
22         if (tablero[0][i] > tablero[1][i]) {
23             buffer = tablero[1][i];
24             tablero[1][i] = tablero[0][i];
25             tablero[0][i] = buffer;
26         }
27         if (tablero[1][i] > tablero[2][i]) {
28             buffer = tablero[2][i];
29             tablero[2][i] = tablero[1][i];
30             tablero[1][i] = buffer;
31         }
32     } while (tablero[0][i] > tablero[1][i] || tablero[1][i] > tablero[2][i] || tablero[0][i] > tablero[2][i]);
33 }
34
35 for (int i = 0; i < tablero.length; i++) {
36     for (int j = 0; j < tablero[i].length; j++) {
37

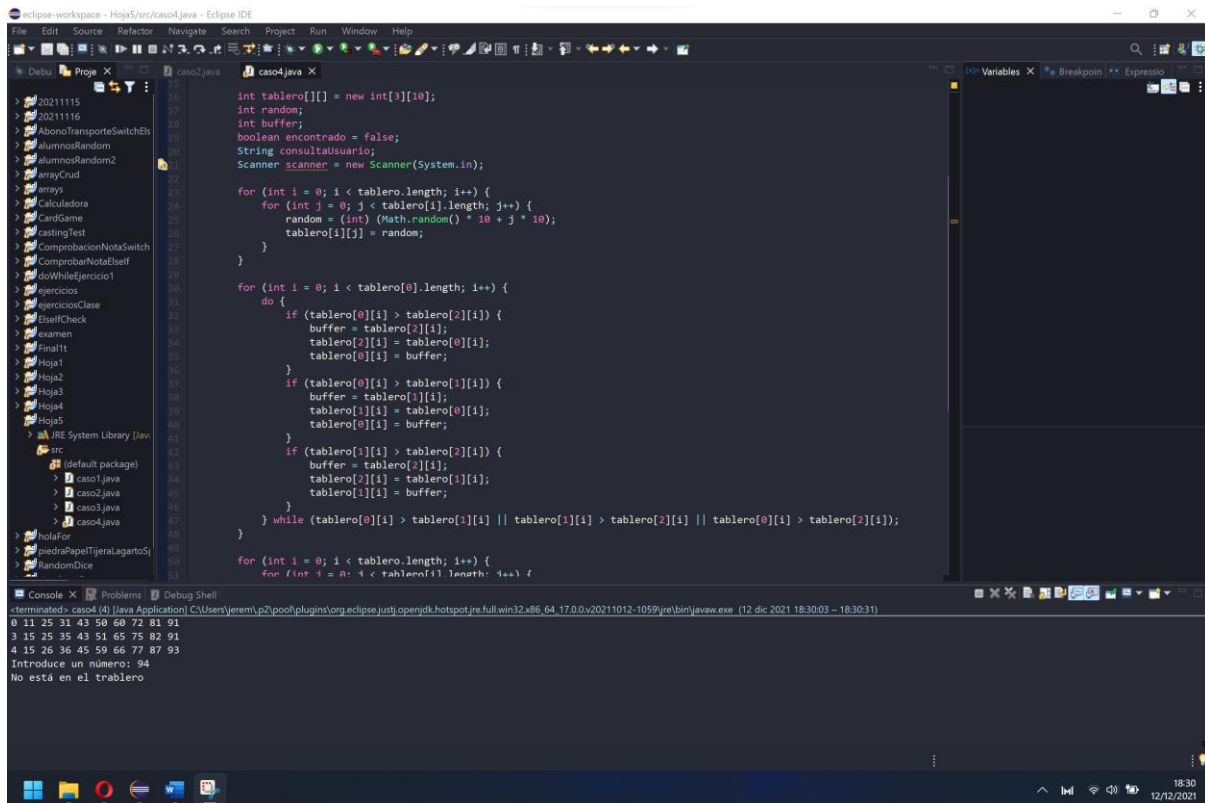
```

Console Output:

```

2 11 22 35 41 56 62 71 83 93
8 18 26 36 45 57 66 75 83 94
8 19 27 37 49 57 67 79 88 96
Introduce un número: 26
Está en el tablero

```



```

15  int tablero[][] = new int[3][10];
16  int random;
17  int buffer;
18  boolean encontrado = false;
19  String consultaUsuario;
20  Scanner scanner = new Scanner(System.in);
21
22  for (int i = 0; i < tablero.length; i++) {
23      for (int j = 0; j < tablero[i].length; j++) {
24          random = (int) (Math.random() * 10 + j * 10);
25          tablero[i][j] = random;
26      }
27  }
28
29  for (int i = 0; i < tablero[0].length; i++) {
30      do {
31          if (tablero[0][i] > tablero[2][i]) {
32              buffer = tablero[2][i];
33              tablero[2][i] = tablero[0][i];
34              tablero[0][i] = buffer;
35          }
36          if (tablero[0][i] > tablero[1][i]) {
37              buffer = tablero[1][i];
38              tablero[1][i] = tablero[0][i];
39              tablero[0][i] = buffer;
40          }
41          if (tablero[1][i] > tablero[2][i]) {
42              buffer = tablero[2][i];
43              tablero[2][i] = tablero[1][i];
44              tablero[1][i] = buffer;
45          }
46          while (tablero[0][i] > tablero[1][i] || tablero[1][i] > tablero[2][i] || tablero[0][i] > tablero[2][i]);
47      } while (true);
48  }
49
50  for (int i = 0; i < tablero.length; i++) {
51      for (int j = 0; j < tablero[i].length; j++) {

```

Console Output:

```

C:\Users\jerem\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full\win32\x86_64\17.0.0.v20211012-1059\jre\bin\java.exe (12 dic 2021 18:30:03 - 18:30:31)
0 11 25 31 43 50 60 72 81 91
3 15 25 35 43 51 65 75 82 91
4 15 26 36 45 59 66 77 87 93
Introduce un número: 94
No está en el tablero

```

Se valora/penaliza:

- Se penalizará si faltan casos de prueba.
- Uso de las variables adecuadas. Justifícalas.
- Uso de las estructuras adecuadas.
- Uso de comentarios útiles.
- Solidez y optimización de la solución.
- Pruebas realizadas. Se deben adjuntar pantallazos de las pruebas de cada caso en este mismo documento, después de cada caso. Si el programa no funciona se debe indicar dónde y la razón.
- El código mal tabulado, el uso de sentencias de interrupción o el uso de materia no vista en clase aún **suponen una nota de cero en el caso práctico**.
- Se entregan los .java y este documento en pdf comprimidos en un archivo zip con “tu nombre – Hoja 4.zip”. El zip debe contener Hoja 4 2021.pdf, caso1.java, caso2. Java, caso3.java, caso4.java y caso5.java. **No se corregirán entregas en otro formato.**