Programación

Act02_05Arrays

Francisco José García Cutillas | 1FPGS_DAM

Índice

Ejercicio A01

Crea un método que devuelva un array a partir de x números enteros aleatorios SIN REPETIR en el rango de dos números enteros que se le pasen como parámetros. Se debe comprobar por supuesto que el parámetro del número de enteros que vaya a generar debe ser mayor que la diferencia entre el menor y el mayor parámetro que se ha pasado para calcular el número aleatorio.

Ejemplo 1: Si se pasa como tamaño de array 10 y el número menor 6 y el mayor 12 devuelve un array vacío (o entero a 0) ya que no es posible generar 10 números sin repetir en dicho rango.
Ejemplo 2: Si se pasa como tamaño de array 10 y el número menor 6 y el mayor 32.

```java
package com.mycompany.garcia_cutillas_franciscojose_act02_05arrays;

/**
 *
 * @author fran
 */
public class Garcia_Cutillas_FranciscoJose_Act02_05Arrays {

    public static void main(String[] args) {

        //Llamada a los métodos de los ejercicios
        Ejercicios ej = new Ejercicios();
        ej.ej01();

    }
}
```

```java
package com.mycompany.garcia_cutillas_franciscojose_act02_05arrays;

import java.util.Arrays;

/**
 *
 * @author fran
 */
public class Ejercicios {

    //Creación de la variable fun para usar la clase Funciones
    Funciones fun = new Funciones();

    public void ej01() {

    //Ejercicio 1
    int resultado[] = fun.creaArrayAleatorio( min: -8, max: 9 ,  longitud: 10);

        System.out.println( x: Arrays.toString( a: resultado));

    }

}
```

```java
package com.mycompany.garcia_cutillas_franciscojose_act02_05arrays;

/**
 *
 * @author fran
 */
public class Funciones {

    public int[] creaArrayAleatorio(int min, int max, int longitud) {

        int arraySalida[] = new int[longitud];
        int numAleatorio = 0;
        boolean arrayCompletd = false;
        int ind = 0;

        //Condiciones de error
        if (min > max) {

            System.out.println( x: "El valor mínimo del array no puede ser mayor que el del máximo");
            arraySalida = null;

        } else if (max - min < longitud - 1) {

            System.out.println("El rango de valores no es suficiente para crear un array "
                    + "sin que se repitan valores");
            arraySalida = null;

         //Proceso si no hay errores
     } else {

         //Genera número aleatorio y lo guardas en el índice 0
         numAleatorio = (int) (Math.random() * ((max - min) + 1)) + min;
         arraySalida[0] = numAleatorio;

         //Proceso de relleno del array para el resto de índices
         for (int i = 1; i < arraySalida.length; i++) {

             boolean diferente = false;

             //Busca número aleatorio y hasta que no sea diferente de los que ya hay no guardes
             do {

                 numAleatorio = (int) (Math.random() * ((max - min) + 1)) + min;

                 for (int j = 0; j < i; j++) {

                     if (arraySalida[j] == numAleatorio) {

                         break;

                     } else if (arraySalida[j] != numAleatorio && (j == i - 1)) {

                         arraySalida[i] = numAleatorio;
                         diferente = true;

                     }

                 }

             } while (!diferente);

         }

         return arraySalida;

    }

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
[0, -5, 4, 7, 2, 9, -8, 6, -3, 3]
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.905 s
Finished at: 2022-12-31T13:11:00+01:00
------------------------------------------------------------------------
```

```
 //Ejercicio 1
 int resultado[] = fun.creaArrayAleatorio( min: 0, max: 10 ,  longitud: 10);

     System.out.println( x: Arrays.toString( a: resultado));

 }
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
[0, 1, 7, 10, 5, 3, 6, 8, 2, 4]
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.949 s
Finished at: 2022-12-31T13:13:14+01:00
------------------------------------------------------------------------
```

```
//Ejercicio 1
int resultado[] = fun.creaArrayAleatorio( min: 10, max: 9 ,  longitud: 10);

    System.out.println( x: Arrays.toString( a: resultado));

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
El valor mínimo del array no puede ser mayor que el del máximo
null
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.979 s
Finished at: 2022-12-31T13:14:34+01:00
```

```
 //Ejercicio 1
 int resultado[] = fun.creaArrayAleatorio( min: 5, max: 9 ,  longitud: 10);

     System.out.println( x: Arrays.toString( a: resultado));

 }
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
El rango de valores no es suficiente para crear un array sin que se repitan valores
null
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.866 s
Finished at: 2022-12-31T13:15:32+01:00
```

Ejercicio A02

Crea un método que devuelva un array a partir de dos números enteros que se le pasen como parámetros. El array que devuelva debe tener

• 20 números aleatorios que no se repitan (Utiliza el método ya creado)
• Un entero con el día del mes en el que estamos. El código para obtenerlo (importando la clase correspondiente) es:
Calendar fecha = Calendar.getInstance();
int dia_actual = fecha.get(Calendar.DAY_OF_MONTH);
• Otro entero con el mes actual.
• Otro entero con el año actual.
• En el main, muestra toda la información.

NOTA: La diferencia entre el número mayor y menor que nos pasen como parámetro debe ser superior a 20 ya que debemos calcular 20 números enteros sin repetición.

```
    public static void main(String[] args) {

        //Llamada a los métodos de los ejercicios
        Ejercicios ej = new Ejercicios();
//          ej.ej01();

        ej.ej02();



    }
}
```

```
//Ejercicio 2
public void ej02() {

    int resultado[] = fun.aleatoriosFecha( min: -10,  max: 9);
    System.out.println( x: Arrays.toString( a: resultado));

}
```

```java
public int[] aleatoriosFecha(int min, int max) {

    int arraySalida[] = new int[23];

    //Creación del array aleatorio de 20 dígitos
    int array20[] = creaArrayAleatorio(min, max, longitud:20);

    //Condición si el método anterior no devuelve error
    if (array20 != null) {

        //Copia del array de 20 aleatorio al array salida
        for (int i = 0; i < array20.length; i++) {

            arraySalida[i] = array20[i];

        }
```

```java
        //Addición de la fecha al array de 23
        Calendar fecha = Calendar.getInstance();
        int dia_actual = fecha.get(field:Calendar.DAY_OF_MONTH);
        int mes_actual = fecha.get(field:Calendar.MONTH);
        int año_actual = fecha.get(field:Calendar.YEAR);

        arraySalida[arraySalida.length - 3] = dia_actual;
        arraySalida[arraySalida.length - 2] = mes_actual + 1;
        arraySalida[arraySalida.length - 1] = año_actual;

        //Si el método devuelve null
    } else {

        arraySalida = null;

    }

    return arraySalida;

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
[6, -6, 4, -3, -9, 8, -4, -5, -8, 2, 5, 9, -1, 1, -2, 0, 7, 3, -7, -10, 2, 1, 2023]
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.875 s
Finished at: 2023-01-02T13:59:10+01:00
------------------------------------------------------------------------
```

```java
//Ejercicio 2
public void ej02() {

    int resultado[] = fun.aleatoriosFecha(min:-10, max:8);
    System.out.println(x:Arrays.toString(a:resultado));

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
El rango de valores no es suficiente para crear un array sin que se repitan valores
null
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.046 s
Finished at: 2023-01-02T14:48:33+01:00
------------------------------------------------------------------------
```

```
//Ejercicio 2
public void ej02() {

    int resultado[] = fun.aleatoriosFecha( min: 10,  max: 8);
    System.out.println( x: Arrays.toString( a: resultado));

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
El valor mínimo del array no puede ser mayor que el del máximo
null
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.028 s
Finished at: 2023-01-02T14:51:21+01:00
```

Ejercicio A03

Crea un método que pida por pantalla el número de personas que quieren comer un día. Para la comida de ese día existen 4 menús y cada una de estas personas deberá introducir su nombre e indicar el menú que quiere comer. Al final de esto el método debe mostrar el menú que ha sido más elegido (en caso de empate se indicará el de menor número), el número de personas que lo han elegido y sus nombres correspondientes.
NOTA: Se puede utilizar más de un array para guardar los datos que se piden por pantalla.

```
public class Garcia_Cutillas_FranciscoJose_Act02_05Arrays {

    public static void main(String[] args) {

        //Llamada a los métodos de los ejercicios
        Ejercicios ej = new Ejercicios();
//        ej.ej01();

//        ej.ej02();

        ej.ej03();

    }
}
```

```java
//Ejercicio 3
public void ej03() {

    fun.menuComida();

}
```

```java
public void menuComida() {

    int numPersonas = 0;

    //Creación variable Scanner para introducir los datos
    Scanner sc = new Scanner( source: System.in);

    //Pregunta cuántas personas van a comer
    System.out.println( x: "¿Cuántas personas van a comer?");
    numPersonas = sc.nextInt();

    //Array en el que se va a guardar los nombres de los comensales
    String nombres[] = new String[numPersonas];

    //Array en el que se va a guardar el menú elegido por cada comensal
    int menu[] = new int[numPersonas];
    int sumaMenu[] = {0, 0, 0, 0};

    //Pregunta nombre y menú elegido por el comensal
    for (int i = 0; i < nombres.length; i++) {

        System.out.println("Introduce el nombre del comensal " + (i + 1) + ":");
        nombres[i] = sc.next();
```

```java
        do {

            System.out.println( x: "Elige menú 1, 2, 3 o 4:");
            menu[i] = sc.nextInt();

            if (menu[i] > 0 && menu[i] <= 4) {

                menuCorrecto = true;

                //Suma la cantidad de menús elegidos
                int auxSuma = menu[i] - 1;
                sumaMenu[auxSuma]++;

            } else {

                System.out.println( x: "Sólo es posible elegir entre el menú 1, 2, 3 o 4");

            }

        } while (!menuCorrecto);
    }
```

```java
        //Cálculo del menú más elegido
        int menuElegido[] = calculaMayor( arrayEntrada: sumaMenu);

        System.out.println("El menú más elegido ha sido el " + menuElegido[1]);
        System.out.println("Ha sido elegido por " + menuElegido[0] + " personas:");
        System.out.println("Las personas que han elegido el menú " + menuElegido[1] + " son:");

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

            if (menu[i] == menuElegido[1]) {

                System.out.println(nombres[i]);

            }

        }

    }
```

```java
//Este método se le introduce un array de enteros y devuelve un array de enteros en el que
//el primer índice indica el mayor y en el segundo el índice en el que se encuentra.
//En caso de empate, devuelve el de índice menor
public int[] calculaMayor(int arrayEntrada[]) {

    int arraySalida[] = new int[2];

    int mayor = arrayEntrada[0];
    int indice = 0;

    //Busca el mayor y guarda su índice. En el caso de encontrar otro número igual,
    //al usar el > y no >= siempre se va a quedar guardado el número de índice menor,
    //aunque encuentre otro igual
    for (int i = 1; i < arrayEntrada.length; i++) {

        if (arrayEntrada[i] > mayor) {

            mayor = arrayEntrada[i];
            indice = i;

        }

    }

    //Rellenado del array de salida
    arraySalida[0] = mayor; //número de personas que eligen el menú
    arraySalida[1] = indice + 1; //Con esto ya tenemos el menú elegido ganador

    return arraySalida;

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
¿Cuántas personas van a comer?
6
Introduce el nombre del comensal 1:
juan
Elige menú 1, 2, 3 o 4:
3
Introduce el nombre del comensal 2:
ana
Elige menú 1, 2, 3 o 4:
1
Introduce el nombre del comensal 3:
loli
Elige menú 1, 2, 3 o 4:
3
Introduce el nombre del comensal 4:
pepe
Elige menú 1, 2, 3 o 4:
4
Introduce el nombre del comensal 5:
antonio
Elige menú 1, 2, 3 o 4:
3
Introduce el nombre del comensal 6:
juana
Elige menú 1, 2, 3 o 4:
2
```

```
El menú más elegido ha sido el 3
Ha sido elegido por 3 personas:
Las personas que han elegido el menú 3 son:
juan
loli
antonio
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  01:01 min
Finished at: 2023-01-02T17:38:25+01:00
------------------------------------------------------------------------
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
¿Cuántas personas van a comer?
4
Introduce el nombre del comensal 1:
antonio
Elige menú 1, 2, 3 o 4:
5
Sólo es posible elegir entre el menú 1, 2, 3 o 4
Elige menú 1, 2, 3 o 4:
0
Sólo es posible elegir entre el menú 1, 2, 3 o 4
Elige menú 1, 2, 3 o 4:
1
Introduce el nombre del comensal 2:
juan
Elige menú 1, 2, 3 o 4:
2
Introduce el nombre del comensal 3:
loli
Elige menú 1, 2, 3 o 4:
2
Introduce el nombre del comensal 4:
pepe
Elige menú 1, 2, 3 o 4:
2
```

```
El menú más elegido ha sido el 2
Ha sido elegido por 3 personas:
Las personas que han elegido el menú 2 son:
juan
loli
pepe
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  51.589 s
Finished at: 2023-01-02T17:44:49+01:00
------------------------------------------------------------------------
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
¿Cuántas personas van a comer?
4
Introduce el nombre del comensal 1:
pepe
Elige menú 1, 2, 3 o 4:
1
Introduce el nombre del comensal 2:
juan
Elige menú 1, 2, 3 o 4:
1
Introduce el nombre del comensal 3:
juana
Elige menú 1, 2, 3 o 4:
3
Introduce el nombre del comensal 4:
ana
Elige menú 1, 2, 3 o 4:
3
El menú más elegido ha sido el 1
Ha sido elegido por 2 personas:
Las personas que han elegido el menú 1 son:
pepe
juan
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  01:26 min
Finished at: 2023-01-02T17:47:32+01:00
------------------------------------------------------------------------
```

Ejercicio A04

Crea un método que devuelva la calificación de un alumno final teniendo en cuenta que:

• El método va a recibir su calificación en el examen (que puede ser un valor con decimales) un array de calificaciones de las actividades (que pueden tener decimales) y un array de ENTEROS con el porcentaje que vale cada una de las actividades (la suma de estos valores debe ser 100).

• Además, va a recibir dos parámetros más: El primero indica la ponderación de lo que vale el examen y el otro el porcentaje de lo que valen las actividades. Deben ser dos valores enteros y la suma de ambos como no debe ser de otra manera debe ser 100.

• Un ejemplo de valores que puede recibir ese método sería: (5.5, {7.2, 6.3, 4.5, 9.5}, {30, 40, 20, 10}, 60, 40). Esto quiere decir que ha sacado un 5.5 en el examen, tiene de nota en cada una de las actividades un 7.2, 6.3, 4.5 y 9.5, valiendo estas tareas un 30%, 40%, 20% y 10% respectivamente. Finalmente, el examen tiene un valor del 60% en el total y las actividades un 40%. Con estos valores el resultado debería ser un 5,912 (ya que las tareas dan como resultado un 6,53 pero valen un 40%).

```java
package com.mycompany.garcia_cutillas_franciscojose_act02_05arrays;

/**
 *
 * @author fran
 */
public class Garcia_Cutillas_FranciscoJose_Act02_05Arrays {

    public static void main(String[] args) {

        //Llamada a los métodos de los ejercicios
        Ejercicios ej = new Ejercicios();
//        ej.ej01();

//        ej.ej02();

//        ej.ej03();

        ej.ej04();
    }
}
```

```java
//Ejercicio 4
public void ej04(){

    double calActividades[] = {7.2, 6.3, 4.5, 9.5};
    int porcActividades[] = {30, 40, 20, 10};

    double resultado = fun.calificacionFinal( calEx: 5.5, calActividades, porcActividades, pondEx: 60, pondAc: 40);

    if (resultado != -1) {

        System.out.println("La calificación final es: "+resultado);

    }


}
```

```java
public double calificacionFinal(double calEx, double calActividades[],
        int porcActividades[], int pondEx, int pondAc) {

    double calFinal = 0.0;

    //Calificación final examen
    double calFinEx = (calEx * ((double) pondEx / 100));

    //Suma ponderaciones totales
    int sumaPondFin = pondEx + pondAc;
    int sumaPondAct = 0;
    boolean errorCalAct = false;

    //Caso en el que la nota del examen sea negativa o mayor de 10
    if (calEx < 0 || calEx > 10) {

        System.out.println( x: "La calificación del examen debe de ser entre 0 y 10");
        calFinal = -1;

    }
```

```
    //Busca si hay algún error en las calificaciones de las actividades y sus ponderaciones
    //Caso en el que no se introduzca el mismo número de calificaciones que de ponderaciones de las mismas
    if (calActividades.length != porcActividades.length) {

        System.out.println("Debes introducir el mismo número de calificaciones de "
                + "actividades, como sus respectivas ponderaciones");
        calFinal = -1;

        //Si no se cumple lo de arriba, comprueba que las calificaciones están entre 0 y 10
    } else {

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

            if (calActividades[i] < 0 || calActividades[i] > 10) {

                System.out.println( x: "La nota de las calificaciones debe ser entre 0 y 10");
                calFinal = -1;
                break;

            }

        }

    }

     //Caso en el que la suma de la ponderación de las actividades sea mayor o menor de 100
     for (int i = 0; i < porcActividades.length; i++) {

         sumaPondAct += porcActividades[i];

     }

     if (sumaPondAct > 100 || sumaPondAct < 100) {

         System.out.println( x: "La suma de la ponderación de las actividades debe ser 100");
         calFinal = -1;

     }

     //Caso en el que la suma de la ponderación final sea mayor o menor de 100
     if (sumaPondFin > 100 || sumaPondFin < 100) {

         System.out.println( x: "La suma de las ponderaciones finales debe ser 100");
         calFinal = -1;

     }

      //Cálculo de la calificación final de las actividades
      if (calFinal != -1) {

          //Notas actividades ya ponderadas
          double calPond[] = new double[calActividades.length];

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

              calPond[i] = calActividades[i] * ((double) porcActividades[i] / 100);

          }

          //Nota final actividades
          double calFinAc = 0.0;

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

              calFinAc += calPond[i];

          }

          //Calificación final examen + actividades
          calFinal = calFinEx + (calFinAc * ((double) pondAc / 100));

      }

      return calFinal;

  }
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
La calificación final es: 5.912
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.866 s
Finished at: 2023-01-03T13:53:09+01:00
```

```java
//Ejercicio 4
public void ej04(){

    double calActividades[] = {6.3, 4.5, 9.5};
    int porcActividades[] = {30, 40, 20, 10};

    double resultado = fun.calificacionFinal( calEx: 5.5, calActividades, porcActividades,  pondEx: 60,  pondAc: 40);

    if (resultado != -1) {

        System.out.println("La calificación final es: "+resultado);

    }


}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
Debes introducir el mismo número de calificaciones de actividades, como sus respectivas ponderaciones
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.878 s
Finished at: 2023-01-03T14:00:45+01:00
------------------------------------------------------------------------
```

```java
//Ejercicio 4
public void ej04(){

    double calActividades[] = {-7.2, 6.3, 4.5, 9.5};
    int porcActividades[] = {30, 40, 20, 10};

    double resultado = fun.calificacionFinal( calEx: 5.5, calActividades, porcActividades,  pondEx: 60,  pondAc: 40);

    if (resultado != -1) {

        System.out.println("La calificación final es: "+resultado);

    }
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
La nota de las calificaciones debe ser entre 0 y 10
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.945 s
Finished at: 2023-01-03T14:02:08+01:00
------------------------------------------------------------------------
```

```java
//Ejercicio 4
public void ej04(){

    double calActividades[] = {7.2, 6.3, 4.5, 9.5};
    int porcActividades[] = {40, 20, 10};

    double resultado = fun.calificacionFinal( calEx: 5.5, calActividades, porcActividades,  pondEx: 60,  pondAc: 40);

    if (resultado != -1) {

        System.out.println("La calificación final es: "+resultado);

    }
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
Debes introducir el mismo número de calificaciones de actividades, como sus respectivas ponderaciones
La suma de la ponderación de las actividades debe ser 100
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.906 s
Finished at: 2023-01-03T14:03:09+01:00
```

```
//Ejercicio 4
public void ej04(){

    double calActividades[] = {7.2, 6.3, 4.5, 9.5};
    int porcActividades[] = {30, 40, 20, 10};

    double resultado = fun.calificacionFinal(-5.5, calActividades, porcActividades,  pondEx: 50,  pondAc: 40);

    if (resultado != -1) {

        System.out.println("La calificación final es: "+resultado);

    }
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
La calificación del examen debe de ser entre 0 y 10
La suma de las ponderaciones finales debe ser 100
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.911 s
Finished at: 2023-01-03T14:04:33+01:00
------------------------------------------------------------------------
```

Ejercicio A05

Implementa un juego en el que se debe simular que participan dos jugadores y cada uno de ellos trata de acertar el mayor número de números que aleatoriamente va a generar una máquina. Una vez que se finalice la partida el programa debe preguntar a los jugadores si desean jugar una partida. Cuestiones sobre el juego:

• El programa le pedirá a cada jugador que indiquen 5 números enteros que se almacenarán en un array que deberán estar en el rango del 1 al 50.

• El programa generará un array de 10 números aleatorios entre el 1 y el 50 sin repetir.

• El programa comprobará cuántos de esos números se encuentran en el array para cada jugador.

• El jugador que más aciertos haya obtenido habrá ganado la partida.

• En caso de empate ganará el jugador que haya introducido los números primero, por lo que una mejora que puede tener el juego es alternar el jugador (son dos jugadores) que mete primero los números.

```java
package com.mycompany.garcia_cutillas_franciscojose_act02_05arrays;

/**
 *
 * @author fran
 */
public class Garcia_Cutillas_FranciscoJose_Act02_05Arrays {

    public static void main(String[] args) {

        //Llamada a los métodos de los ejercicios
        Ejercicios ej = new Ejercicios();
//        ej.ej01();

//        ej.ej02();

//        ej.ej03();

//        ej.ej04();

        ej.ej05();
    }
}
```

```java
//Ejercicio 5
public void ej05(){

    fun.juegoAciertaNum();

}
```

```java
public void juegoAciertaNum() {

    boolean finJuego = false;
    int contadorJuego = 0;

    //El juego se ejecutará hasta que los jugadores decidan no seguir jugando
    do {

        //Genera array aleatorio de longitud 10 sin repeticiones
        int numAdivinar[] = creaArrayAleatorio( min: 1,  max: 50,  longitud: 10);

        //Condición para preguntar primero a un jugador u otro
        boolean jugPrincipal = true;//Si está a true el principal es el jugador 1
        int jugador1[] = new int[5];
        int jugador2[] = new int[5];

        if (contadorJuego % 2 == 0) {

            System.out.println( x: "****Jugador 1****");
            jugador1 = pide5Num();

            System.out.println( x: "****Jugador 2****");
            jugador2 = pide5Num();

            contadorJuego++;
```

```
        } else {

            jugPrincipal = false;//Jugador principal el jugador 2

            System.out.println( x: "****Jugador 2****");
            jugador2 = pide5Num();

            System.out.println( x: "****Jugador 1****");
            jugador1 = pide5Num();

            contadorJuego++;

        }

        //Muestra arrays de jugadores y el aleatorio
        System.out.println("jugador 1 ->" + Arrays.toString( a: jugador1));
        System.out.println("jugador 2 ->" + Arrays.toString( a: jugador2));
        System.out.println("Array generado ->" + Arrays.toString( a: numAdivinar));

        //Comprobar aciertos de cada jugador
        int acJugador1 = 0;
        int acJugador2 = 0;

        acJugador1 = compruebaCoincidencias( array1: jugador1,  array2: numAdivinar);
        acJugador2 = compruebaCoincidencias( array1: jugador2,  array2: numAdivinar);

        System.out.println("Aciertos jugador 1: "+ acJugador1);
        System.out.println("Aciertos jugador 2: "+ acJugador2);
```

```
        //Determina el jugador ganador
        if (acJugador1 > acJugador2) {

            System.out.println( x: "Ganador -> jugador 1");

        } else if (acJugador1 < acJugador2) {

            System.out.println( x: "Ganador -> jugador 2");

            //En caso de empate, gana el jugador principal
        } else if (acJugador1 == acJugador2) {

            if (acJugador1 == 0) {

                System.out.println( x: "Ningún jugador ha acertado");

            } else if (jugPrincipal) {

                System.out.println( x: "Ganador jugador -> 1");

            } else if (!jugPrincipal) {

                System.out.println( x: "Ganador jugador -> 2");

            }

        }
```

```
        //Condición para finalizar el juego
        String jugar = "";

        do {

            System.out.println( x: "¿Deseas seguir jugando? (s/n)");
            Scanner sc = new Scanner( source: System.in);
            jugar = sc.next();

            //Sólo se permite contestar con s/n, de lo contrario indica que no es correcto
            //y vuelve a preguntar hasta que se escriba correctamente
            if (jugar.equals( anObject: "n") || jugar.equals( anObject: "N")) {

                finJuego = true;
                break;

            } else if (jugar.equals( anObject: "S") || jugar.equals( anObject: "s")) {

                break;

            } else {

                System.out.println( x: "Sólo se permite contestar con s/n");

            }

        } while (!"n".equals( anObject: jugar) || !"N".equals( anObject: jugar)
                || !"S".equals( anObject: jugar) || !"s".equals( anObject: jugar));

    } while (!finJuego);

}

//Este método te pide 5 números en un rango del 1 al 50 y los devuelve en un array
public int[] pide5Num() {

    int arraySalida[] = new int[5];
    Scanner sc = new Scanner( source: System.in);

    System.out.println( x: "Introduce 5 números enteros del 1 al 50");

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

        boolean numCorrecto = false;

        do {

            System.out.println("Número " + (i + 1));
            arraySalida[i] = sc.nextInt();

            if (arraySalida[i] > 0 && arraySalida[i] <= 50) {

                numCorrecto = true;
```

```
            } else {

                System.out.println( x: "Debes introducir un número del 1 al 50");

            }

        } while (!numCorrecto);

    }

    return arraySalida;
}

//Este método nos permite saber si los números del array 1 están contenidos en el array 2
//y devuelve los aciertos
public int compruebaCoincidencias(int array1[], int array2[]) {

    int aciertos = 0;

    //Recorre array del jugador
    for (int i = 0; i < array1.length; i++) {

        //Recorre array aleatorio buscando coincidencias
        for (int j = 0; j < array2.length; j++) {

            if (array1[i] == array2[j]) {

                aciertos++;//Si hay alguna coincidencia, añade acierto al jugador

            }

        }

    }

    return aciertos;

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
****Jugador 1****
Introduce 5 números enteros del 1 al 50
Número 1
1
Número 2
2
Número 3
3
Número 4
4
Número 5
5
****Jugador 2****
Introduce 5 números enteros del 1 al 50
Número 1
6
Número 2
7
Número 3
8
Número 4
9
Número 5
10
```

```
jugador 1 ->[1, 2, 3, 4, 5]
jugador 2 ->[6, 7, 8, 9, 10]
Array generado ->[10, 17, 47, 40, 26, 29, 3, 24, 50, 43]
Aciertos jugador 1: 1
Aciertos jugador 2: 1
Ganador jugador -> 1
¿Deseas seguir jugando? (s/n)
s
****Jugador 2****
Introduce 5 números enteros del 1 al 50
Número 1
1
Número 2
2
Número 3
3
Número 4
4
Número 5
5
****Jugador 1****
Introduce 5 números enteros del 1 al 50
Número 1
6
Número 2
7
Número 3
8
Número 4
9
Número 5
10
jugador 1 ->[6, 7, 8, 9, 10]
jugador 2 ->[1, 2, 3, 4, 5]
Array generado ->[39, 26, 11, 19, 14, 18, 12, 17, 30, 36]
Aciertos jugador 1: 0
Aciertos jugador 2: 0
Ningún jugador ha acertado
¿Deseas seguir jugando? (s/n)
s
****Jugador 1****
Introduce 5 números enteros del 1 al 50
Número 1
5
Número 2
6
Número 3
8
Número 4
15
Número 5
22
****Jugador 2****
Introduce 5 números enteros del 1 al 50
Número 1
26
Número 2
5
Número 3
74
Debes introducir un número del 1 al 50
Número 3
33
```

```
Número 4
26
Número 5
30
jugador 1 ->[5, 6, 8, 15, 22]
jugador 2 ->[26, 5, 33, 26, 30]
Array generado ->[37, 42, 1, 44, 14, 34, 28, 33, 3, 47]
Aciertos jugador 1: 0
Aciertos jugador 2: 1
Ganador -> jugador 2
¿Deseas seguir jugando? (s/n)
n
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  03:09 min
Finished at: 2023-01-03T20:07:48+01:00
```

Ejercicio A06

Implementa el juego del ahorcado. Para ello vamos a obtener un array de char a partir de un array de String que tengamos previamente guardado (ya creamos un método que hacía esta conversión utilizando el método toCharArray de la clase String).
El usuario tiene 7 intentos para acertar la palabra que se ha elegido (utiliza el método anterior para seleccionar la palabra). El programa debe ir pidiendo letras al usuario y tras cada intento mostrar si se encuentra o no para reducir los intentos y mostrar. El estado en el que se encuentra su "adivinación". Al finalizar indicará si ha ganado o ha perdido porque ha agotado los intentos. Tras finalizar una partida el programa debe preguntar si quiere seguir jugando una nueva partida (calcula una nueva palabra) o si por el contrario quiere finalizar.

```
package com.mycompany.garcia_cutillas_franciscojose_act02_05arrays;

/**
 *
 * @author fran
 */
public class Garcia_Cutillas_FranciscoJose_Act02_05Arrays {

    public static void main(String[] args) {

        //Llamada a los métodos de los ejercicios
        Ejercicios ej = new Ejercicios();
//        ej.ej01();

//        ej.ej02();

//        ej.ej03();

//        ej.ej04();

//        ej.ej05();

        ej.ej06();

    }
}
```

```
public void ej06() {

    fun.ahorcado();

}
```

```java
public void ahorcado() {

    Scanner sc = new Scanner( source: System.in);

    String entrada[] = {"hola", "programacion", "sistemas", "aula", "juan"};

    boolean juego = true;

    System.out.println( x: "******Juego del ahorcado******");

    while (juego) {

        //Genera palabra aleatoria
        char palabraAleatoria[] = convierteStringChar(entrada);

        //Encripta la palabra
        char palabraEncriptada[] = encriptaArray( entrada: palabraAleatoria);

        //Tenemos 7 intentos para adivinar y el juego termina cuando el usuario decida
        int intentos = 0;

        //Variable para saber cuándo está la palabra descubierta del todo
        int contadorLetra = cuentaLetras( entrada: palabraEncriptada);

        //Muestra la pista
        System.out.println( x: "Pista de la palabra que tienes que adivinar");
        System.out.println( x: Arrays.toString( a: palabraEncriptada));

        do {

            //Pregunta una letra para ver si se encuentra y búscala
            System.out.println( x: "Introduce una letra para ver si se encuentra en la palabra:");
            char letraIntroducida = sc.next().charAt( index: 0);

            boolean letraEncontrada = false;//Variable para determinar si se encuentra la letra en la palabra

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

                //Si encuentra la letra, colócala en el array resultado
                if (letraIntroducida == palabraAleatoria[i]) {

                    palabraEncriptada[i] = letraIntroducida;
                    letraEncontrada = true;

                }

            }

            //Muestra cómo quedaría el array después y resta uno al contador de letras por descubrir
            if (letraEncontrada) {

                System.out.println("La letra introducida se encuentra en la palabra. "
                        + "Te muestro el resultado:");
                System.out.println( x: Arrays.toString( a: palabraEncriptada));
                contadorLetra = cuentaLetras( entrada: palabraEncriptada);//Calcula las letras que quedan

              //Si no se encuentra la letra, suma un intento y muestra gráficamente
            } else {

                intentos++;
                System.out.println("La letra introducida no se encuentra en la palabra. "
                        + "Te quedan " + (7 - intentos) + " intentos.");
                muñeco(intentos);

            }

            //Palabra adivinada. Se acaba el juego
            if (contadorLetra == 0) {

                System.out.println( x: "Has adivinado la palabra");
                break;

            }

        } while (intentos < 7 || contadorLetra == 0);

        //Condición para seguir jugando cuando se adivine la palabra o se acaben los intentos
        boolean respuestaCorrecta = false;
```

```
        do {

            System.out.println( x: "¿Quieres seguir jugando? s/n");

            char respuesta = sc.next().charAt( index: 0);

            switch (respuesta) {

                case 'n', 'N' -> {
                    System.out.println( x: "Fin del juego");
                    juego = false;
                    respuestaCorrecta = true;
                }

                case 's', 'S' -> {
                    System.out.println( x: "**Nueva palabra**");
                    respuestaCorrecta = true;
                }

                default ->
                    System.out.println( x: "Debes elegir sólo s/n");
            }

        } while (!respuestaCorrecta);

    }

}
```

```
//Este método recibe un array de String y devuelve una de esas palabras aletatoriamente en forma de
//array de char
public char[] convierteStringChar(String entrada[]) {

    int palabraAleatoria = (int) (Math.random() * (((entrada.length - 1) - 0) + 1));

    char[] res = entrada[palabraAleatoria].toCharArray();

    return res;

}
//Método para encriptar un array de char salvo primera y última letra
//si alguna letra de la palabra contiene la primera o la última también es mostrada
public char[] encriptaArray(char entrada[]) {

    char arraySalida[] = new char[entrada.length];

    //Guarda primera y última letra para buscar coincidencias en el array
    arraySalida[0] = entrada[0];
    arraySalida[entrada.length - 1] = entrada[entrada.length - 1];

    //Recorre el array y si encuentras una letra igual que la primera o la última,
    //la muestras. De lo contrario la encriptas
    for (int i = 1; i < entrada.length - 1; i++) {

        if (entrada[i] == arraySalida[0] || entrada[i] == arraySalida[entrada.length - 1]) {

            arraySalida[i] = entrada[i];

        } else {

            arraySalida[i] = '*';

        }

    }

    return arraySalida;
```

```java
//Método para contar las letras que hay que adivinar
public int cuentaLetras(char entrada[]) {

    int letras = 0;

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

        if (entrada[i] == '*') {

            letras++;

        }

    }

    return letras;

}
//Método para mostrar gráficamente el muñeco del ahorcado
public void muñeco(int intentos) {

    switch (intentos) {

        case 0:
            System.out.println("***************");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*******");
            break;

       case 1:
            System.out.println("***************");
            System.out.println("**              *");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*******");
            break;

       case 2:
            System.out.println("***************");
            System.out.println("**              *");
            System.out.println("**              *");
            System.out.println("**            *   *");
            System.out.println("**              *");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*");
            System.out.println("*******");
            break;
```

```
        case 3:
            System.out.println( x: "***************");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**           *   *");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**");
            System.out.println( x: "**");
            System.out.println( x: "**");
            System.out.println( x: "******");
            break;

        case 4:
            System.out.println( x: "***************");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**           *   *");
            System.out.println( x: "**             *");
            System.out.println( x: "**           ***");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**");
            System.out.println( x: "**");
            System.out.println( x: "**");
            System.out.println( x: "******");
            break;
```

```
        case 5:
            System.out.println( x: "***************");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**           *   *");
            System.out.println( x: "**             *");
            System.out.println( x: "**           *****");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**");
            System.out.println( x: "**");
            System.out.println( x: "**");
            System.out.println( x: "******");
            break;

        case 6:
            System.out.println( x: "***************");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**           *   *");
            System.out.println( x: "**             *");
            System.out.println( x: "**           *****");
            System.out.println( x: "**             *");
            System.out.println( x: "**             *");
            System.out.println( x: "**            *");
            System.out.println( x: "**           *");
            System.out.println( x: "**");
            System.out.println( x: "******");
            break;
```

```java
        case 7:
            System.out.println( x: "***************");
            System.out.println( x: "**              *");
            System.out.println( x: "**              *");
            System.out.println( x: "**            *   *");
            System.out.println( x: "**              *");
            System.out.println( x: "**            *****");
            System.out.println( x: "**              *");
            System.out.println( x: "**              *");
            System.out.println( x: "**             * *");
            System.out.println( x: "**            *   *");
            System.out.println( x: "*");
            System.out.println( x: "******      Muerto");
            break;

    }

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
******Juego del ahorcado******
Pista de la palabra que tienes que adivinar
[s, *, s, *, *, *, *, s]
Introduce una letra para ver si se encuentra en la palabra:
l
La letra introducida no se encuentra en la palabra. Te quedan 6 intentos.
***************
*             *
*
*
*
*
*
*
*
*
*
******
Introduce una letra para ver si se encuentra en la palabra:
i
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[s, i, s, *, *, *, *, s]
Introduce una letra para ver si se encuentra en la palabra:
e
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[s, i, s, *, e, *, *, s]
Introduce una letra para ver si se encuentra en la palabra:
o
La letra introducida no se encuentra en la palabra. Te quedan 5 intentos.
****************
*              *
*              *
*            *   *
*              *
*
*
*
*
*
*
******
Introduce una letra para ver si se encuentra en la palabra:
t
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[s, i, s, t, e, *, *, s]
Introduce una letra para ver si se encuentra en la palabra:
a
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[s, i, s, t, e, *, a, s]
```

```
Introduce una letra para ver si se encuentra en la palabra:
r
La letra introducida no se encuentra en la palabra. Te quedan 4 intentos.
***************
*              *
*              *
*            *   *
*              *
*              *
*              *
*              *
*
*
*
******
Introduce una letra para ver si se encuentra en la palabra:
m
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[s, i, s, t, e, m, a, s]
Has adivinado la palabra
¿Quieres seguir jugando? s/n
s
**Nueva palabra**
Pista de la palabra que tienes que adivinar
[p, *, *, *, *, *, *, *, *, *, *, n]
Introduce una letra para ver si se encuentra en la palabra:
m
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[p, *, *, *, *, *, m, *, *, *, *, n]
Introduce una letra para ver si se encuentra en la palabra:
i
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[p, *, *, *, *, *, m, *, *, i, *, n]
Introduce una letra para ver si se encuentra en la palabra:
t
La letra introducida no se encuentra en la palabra. Te quedan 6 intentos.
***************
*             *
*
*
*
*
*
*
*
*
*
******
Introduce una letra para ver si se encuentra en la palabra:
r
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[p, r, *, *, r, *, m, *, *, i, *, n]
Introduce una letra para ver si se encuentra en la palabra:
g
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[p, r, *, g, r, *, m, *, *, i, *, n]
Introduce una letra para ver si se encuentra en la palabra:
o
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[p, r, o, g, r, *, m, *, *, i, o, n]
```

```
Introduce una letra para ver si se encuentra en la palabra:
a
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[p, r, o, g, r, a, m, a, *, i, o, n]
Introduce una letra para ver si se encuentra en la palabra:
c
La letra introducida se encuentra en la palabra. Te muestro el resultado:
[p, r, o, g, r, a, m, a, c, i, o, n]
Has adivinado la palabra
¿Quieres seguir jugando? s/n
n
Fin del juego
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  01:39 min
Finished at: 2023-01-04T18:54:04+01:00
------------------------------------------------------------------------
```

Ejercicio A07

Implementa un método que reciba dos arrays de lo que queramos (char o int por ejemplo) e indique si el segundo está contenido (devuelve un booleano), pudiendo estar este contenido tanto en un sentido como en el otro (pequeña sopa de letras en la que queremos encontrar una palabra, pero la sopa de letras solamente tiene una fila).

Ejemplo:

A	B	W	E	T	L	L	O	M	N

Si el segundo array es T, E, W deberá devolver TRUE ya que está en la posición 4, 3 y 2 del array, así como devolverá TRUE si el segundo array es O, M, N ya que se encuentran en la posición 7, 8 y 9.
Sin embargo, devolverá FALSE si introduce A, B y E ya que, aunque se encuentran los tres caracteres no están de manera consecutiva según el array original tanto si leemos hacia la derecha como hacia a la izquierda en el mismo.

```java
package com.mycompany.garcia_cutillas_franciscojose_act02_05arrays;

/**
 *
 * @author fran
 */
public class Garcia_Cutillas_FranciscoJose_Act02_05Arrays {

    public static void main(String[] args) {

        //Llamada a los métodos de los ejercicios
        Ejercicios ej = new Ejercicios();
//        ej.ej01();

//        ej.ej02();

//        ej.ej03();

//        ej.ej04();

//        ej.ej05();

//        ej.ej06();

        ej.ej07();

    }
}
```

```java
public void ej07() {

    char array1[] = {'a', 'b', 'c', 'd', 'e'};
    char array2[] = {'b', 'a', 'c'};

    //Si el array principal es de menor longitud que el secundario
    if (array2.length > array1.length) {

        System.out.println( x: "El array principal no puede ser de menor tamaño que el secundario");

    } else {

        boolean res = fun.arrayContenido(array1, array2);
        System.out.println("El array " + Arrays.toString( a: array2) + " está contenido en el "
                + "array " + Arrays.toString( a: array1) + ": " + res);

    }

}
```

```java
//Este método devuelve true si el array2 está contenido en el 1 en ambos sentidos
public boolean arrayContenido(char array1[], char array2[]) {

    boolean salida = false;

    boolean derechaIzquierda = compruebaArray(array1, array2);
    boolean izquierdaDerecha = compruebaArrayInverso(array1, array2);

    //Devuelve true si cualquiera de las dos condiciones es true
    if (derechaIzquierda || izquierdaDerecha) {
        salida = true;
    }

    return salida;

}
```

```java
//Este método recorre el array1 de izquierda a derecha y comprueba si el array2 está contenido
public boolean compruebaArray(char array1[], char array2[]) {

    boolean salida = false;

    //Recorre el array1 en busca de la primera coincidencia con el elemento 0 del array2
    //Busca hasta que exista la posibilidad de que el array2 esté contenido en el array1
    for (int i = 0; i < array1.length - (array2.length - 1); i++) {

        //Caso en el que encuentra coincidencia en el primer índice del array2
        if (array1[i] == array2[0]) {

            //Recorre array secundario y comprueba que los siguientes están en el principal
            for (int j = 1; j < array2.length; j++) {

                if (array1[i + j] != array2[j]) {

                    salida = false;

                } else {

                    salida = true;

                }
            }

        }

    }

    return salida;

}
//Este método recorre el array1 de derecha a izquierda y comprueba si está contenido el array2
public boolean compruebaArrayInverso(char array1[], char array2[]) {

    boolean salida = false;

    //Recorre el array1 en sentido inverso en busca de la primera coincidencia con el elemento 0 del array2
    //Busca hasta que exista la posibilidad de que el array2 esté contenido en el array1
    for (int i = array1.length - 1; i >= array2.length - 1; i--) {

        //Caso en el que encuentra coincidencia en el primer índice del array2
        if (array1[i] == array2[0]) {

            //Recorre array secundario y comprueba que los siguientes están en el principal
            for (int j = 1; j < array2.length; j++) {

                if (array1[i - j] != array2[j]) {

                    salida = false;

                } else {

                    salida = true;

                }

            }

        }

    }

    return salida;

}
```

```java
public void ej07() {

    char array1[] = {'a', 'b', 'c', 'd', 'e'};
    char array2[] = {'b', 'a', 'c'};

    //Si el array principal es de menor longitud que el secundario
    if (array2.length > array1.length) {

        System.out.println( x: "El array principal no puede ser de menor tamaño que el secundario");

    } else {

        boolean res = fun.arrayContenido(array1, array2);
        System.out.println("El array " + Arrays.toString( a: array2) + " está contenido en el "
                + "array " + Arrays.toString( a: array1) + ": " + res);

    }

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
El array [b, a, c] está contenido en el array [a, b, c, d, e]: false
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.925 s
Finished at: 2023-01-04T20:20:40+01:00
```

```java
public void ej07() {

    char array1[] = {'a', 'b', 'c', 'd', 'e'};
    char array2[] = {'d', 'e', 'f'};

    //Si el array principal es de menor longitud que el secundario
    if (array2.length > array1.length) {

        System.out.println( x: "El array principal no puede ser de menor tamaño que el secundario");

    } else {

        boolean res = fun.arrayContenido(array1, array2);
        System.out.println("El array " + Arrays.toString( a: array2) + " está contenido en el "
                + "array " + Arrays.toString( a: array1) + ": " + res);

    }

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
El array [d, e, f] está contenido en el array [a, b, c, d, e]: false
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.979 s
Finished at: 2023-01-04T20:19:32+01:00
```

```java
public void ej07() {

    char array1[] = {'a', 'b', 'c', 'd', 'e'};
    char array2[] = {'b', 'c', 'd'};

    //Si el array principal es de menor longitud que el secundario
    if (array2.length > array1.length) {

        System.out.println( x: "El array principal no puede ser de menor tamaño que el secundario");

    } else {

        boolean res = fun.arrayContenido(array1, array2);
        System.out.println("El array " + Arrays.toString( a: array2) + " está contenido en el "
                + "array " + Arrays.toString( a: array1) + ": " + res);

    }

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
El array [b, c, d] está contenido en el array [a, b, c, d, e]: true
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.939 s
Finished at: 2023-01-04T20:18:36+01:00
```

```
public void ej07() {

    char array1[] = {'a', 'b', 'c', 'd', 'e'};
    char array2[] = {'c', 'b', 'a'};

    //Si el array principal es de menor longitud que el secundario
    if (array2.length > array1.length) {

        System.out.println( x: "El array principal no puede ser de menor tamaño que el secundario");

    } else {

        boolean res = fun.arrayContenido(array1, array2);
        System.out.println("El array " + Arrays.toString( a: array2) + " está contenido en el "
                + "array " + Arrays.toString( a: array1) + ": " + res);

    }

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
El array [c, b, a] está contenido en el array [a, b, c, d, e]: true
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.941 s
Finished at: 2023-01-04T20:16:49+01:00
------------------------------------------------------------------------
```

```
public void ej07() {

    char array1[] = {'a', 'b', 'c', 'd', 'e'};
    char array2[] = {'c', 'b', 'a', 'd', 'r', 't'};

    //Si el array principal es de menor longitud que el secundario
    if (array2.length > array1.length) {

        System.out.println( x: "El array principal no puede ser de menor tamaño que el secundario");

    } else {

        boolean res = fun.arrayContenido(array1, array2);
        System.out.println("El array " + Arrays.toString( a: array2) + " está contenido en el "
                + "array " + Arrays.toString( a: array1) + ": " + res);

    }

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Garcia_Cutillas_FranciscoJose_Act02_05Arrays ---
El array principal no puede ser de menor tamaño que el secundario
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.951 s
Finished at: 2023-01-04T20:13:34+01:00
------------------------------------------------------------------------
```