

EJERCICIOS ARRAYS

1. Crea un array de 10 posiciones de números con valores pedidos por teclado. Muestra por consola el índice y el valor al que corresponde. Haz dos métodos, uno para rellenar valores y otro para mostrar.

```
import javax.swing.JOptionPane;
public class arrayApp {

    public static void main(String[] args) {

        //Esto es opcional
        final int TAMANIO=10;

        int num[]=new int[TAMANIO];

        //Invocamos las funciones
        rellenarArray(num);

        mostrarArray(num);
    }

    public static void rellenarArray(int lista[]){
        for(int i=0;i<lista.length;i++){
            String texto=JOptionPane.showInputDialog("Introduce un número");
            lista[i]=Integer.parseInt(texto);
        }
    }

    public static void mostrarArray(int lista[]){
        for(int i=0;i<lista.length;i++){
            System.out.println("En el índice "+i+" esta el valor "+lista[i]);
        }
    }
}
```

2. Crea un array de números donde le indicamos por teclado el tamaño del array, rellenaremos el array con números aleatorios entre 0 y 9, al final muestra por pantalla el valor de cada posición y la suma de todos los valores. Haz un método para rellenar el array (que tenga como parámetros los números entre los que tenga que generar), para mostrar el contenido y la suma del array y un método privado para generar número aleatorio

```

import javax.swing.JOptionPane;
public class arrayNumAleatoriosApp {

    public static void main(String[] args) {

        //Indicamos el tamaño
        String texto=JOptionPane.showInputDialog("Introduce un tamaño");
        int num[]=new int[Integer.parseInt(texto)];

        //Invocamos las funciones
        rellenarNumAleatorioArray(num, 0, 9);

        mostrarArray(num);
    }

    public static void rellenarNumAleatorioArray(int lista[], int a, int b){
        for(int i=0;i<lista.length;i++){
            //Generamos un número entre los parametros pasados
            lista[i]=((int)Math.floor(Math.random()*(a-b)+b));
        }
    }

    public static void mostrarArray(int lista[]){
        for(int i=0;i<lista.length;i++){
            System.out.println("En el índice "+i+" esta el valor "+lista[i]);
        }
    }
}

```

3. Crea un array de números de un tamaño pasado por teclado, el array contendrá números aleatorios primos entre los números deseados, por último nos indica cual es el mayor de todos.

Haz un método para comprobar que el número aleatorio es primo, puedes hacer todos lo métodos que necesites.

```

import javax.swing.JOptionPane;

public class Ejercicio_DiscoDurodeRoer_Arrays_3 {

    public static void main(String[] args) {

        //Indicamos el tamaño
        String texto=JOptionPane.showInputDialog("Introduce un tamaño");
        int num[]=new int[Integer.parseInt(texto)];

        //Invocamos las funciones
        rellenarNumPrimosAleatorioArray(num, 1, 100);

        mostrarArray(num);

        //Saco el primo mayor
        int primoMayor=mayor(num);
        System.out.println("El primo más grande es el "+primoMayor);
    }

    public static void rellenarNumPrimosAleatorioArray(int lista[], int a, int b){

        int i=0;

        //Usamos mejor un while, ya que solo aumentara cuando genere un primo
        while(i<lista.length){
            int num=((int)Math.floor(Math.random()*(a-b)+b));
            if (esPrimo(num)){
                lista[i]=num;
                i++;
            }
        }

    }

    private static boolean esPrimo (int num){

        //Un numero negativo, el 0 y el 1, son directamente no primos.
        if (num<=1){
            return false;
        }else{

            //declaracion
            int prueba;
            int contador=0;
            //Hacemos la raiz cuadrada y lo usamos para dividir el numero original
            prueba=(int)Math.sqrt(num);
            //Bucle que cuenta los numeros divisibles, podemos hacerlo con while
            for (;prueba>1;prueba--){
                if (num%prueba==0){
                    contador+=1;
                }
            }
            return contador < 1;
        }
    }

    public static void mostrarArray(int lista[]){
        for(int i=0;i<lista.length;i++){
            System.out.println("En el indice "+i+" esta el valor "+lista[i]);
        }
    }

    public static int mayor(int lista[]){
        int mayor=0;
        for(int i=0;i<lista.length;i++){

```

4. Crea un array de números de 100 posiciones, que contendrá los números del 1 al 100. Obtén la suma de todos ellos y la media.

```
public class SumaMediaArrayApp {  
  
    public static void main(String[] args) {  
  
        //Creo un array  
        int[] numeros=new int[100];  
  
        //Declaro las variables necesarias  
        int suma=0;  
        double media;  
  
        //Recorro el array, asigno números y sumo  
        for(int i=0;i<numeros.length;i++){  
            numeros[i]=i+1;  
            suma+=numeros[i];  
        }  
  
        //Calculo la media y muestro la suma y la media  
        System.out.println("La suma es "+suma);  
  
        media=(double)suma/numeros.length;  
  
        System.out.println("La media es "+media);  
  
    }  
}
```

5. Crea un array de caracteres que contenga de la 'A' a la 'Z' (solo las mayúsculas). Después, ve pidiendo posiciones del array por teclado y si la posición es correcta, se añadirá a una cadena que se mostrara al final, se dejará de insertar cuando se introduzca un -1.

```

import java.util.Scanner;

public class Ejercicio_DiscoDurodeRoer_Arrays_5 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        char mayusculas[] = new char[26];
        for (int i = 65, j = 0; i <= 90; i++, j++) {
            mayusculas[j] = (char) i;
        }

        String cadena = "";
        int eleccion = -1;

        do {
            System.out.println("Elija un indice entre 0 y " + (mayusculas.length - 1));
            eleccion = sc.nextInt();

            if (!(eleccion >= 0 && eleccion <= mayusculas.length - 1)) {
                System.out.println("Error, inserte otro numero");
            } else {
                if (eleccion != -1) {
                    cadena += mayusculas[eleccion];
                }
            }

        } while (eleccion != -1);

        System.out.println(cadena);

    }

}

```

Por ejemplo, si escribo los siguientes numeros

```

0 //Añadira la 'A'
5 //Añadira la 'F'
25 //Añadira la 'Z'
50 //Error, inserte otro numero
-1 //fin
Cadena resultante: AFZ

```

6. Pide al usuario por teclado una frase y pasa sus caracteres a un array de caracteres. Puedes hacer con o sin métodos de String.

1: sin metodos de String

```
import java.util.Scanner;

public class MuestraCaracteresApp {

    public static void main(String[] args) {

        Scanner sn=new Scanner(System.in);

        //Pido al usuario que escriba una frase
        System.out.println("Escriba una frase");

        //Recomiendo usar nextLine en lugar de next por los espacios
        String frase=sn.nextLine();

        //Creamos un array de caracteres
        char caracteres[]=new char[frase.length()];

        //Recorremos la frase y cogemos cada caracter y lo metemos en el array
        for(int i=0;i<frase.length();i++){
            caracteres[i]=frase.charAt(i);
            System.out.println(caracteres[i]);
        }

    }

}
```

2: con metodos String

```
import java.util.Scanner;

public class MuestraCaracteresApp {

    public static void main(String[] args) {

        Scanner sn=new Scanner(System.in);

        //Pido al usuario que escriba una frase
        System.out.println("Escriba una frase");

        //Recomiendo usar nextLine en lugar de next por los espacios
        String frase=sn.nextLine();

        //Creamos un array de caracteres usando el metodo de String
        char caracteres[]=frase.toCharArray();

        //Recorremos la frase y cogemos cada caracter y lo metemos en el array
        for(int i=0;i<frase.length();i++){
            System.out.println(caracteres[i]);
        }

    }

}
```


7. Crea dos arrays de números con una posición pasado por teclado.

Uno de ellos estará relleno con números aleatorios y el otro apuntara al array anterior, después crea un nuevo array con el primer array (usa de nuevo new con el primer array) con el mismo tamaño que se ha pasado por teclado, rellenalos de nuevo con números aleatorios.

Después, crea un método que tenga como parámetros, los dos arrays y devuelva uno nuevo con la multiplicación de la posición 0 del array1 con el del array2 y así sucesivamente. Por último, muestra el contenido de cada array.

Llama al final al recolector de basura.

```
import javax.swing.JOptionPane;
public class MultiplicacionApp {

    public static void main(String[] args) {

        //Indicamos el tamaño
        String texto=JOptionPane.showInputDialog("Introduce un tamaño");
        int tamaño=Integer.parseInt(texto);

        //Creamos los arrays
        int array1[]=new int [tamaño];
        int array2[];

        //Rellenamos el array1
        rellenarNumAleatorioArray(array1, 10, 100);

        //Apuntamos el array2 al array1
        array2=array1;

        //Creamos un nuevo array, usando el array1. Array1 tendra una nueva direccion.
        array1=new int[tamaño];

        //Lo volvemos a rellenar, ya que al crear el array de nuevo no contiene los numeros anteriores
        rellenarNumAleatorioArray(array1, 10, 100);

        //Contiene el array con el resultado de multiplicar los valores de los arrays
        int array3[]=multiplicador(array1, array2);

        //Mostramos el contenido de los arrays

        System.out.println("Array1");
        mostrarArray(array1);
```

```

        System.out.println("Array2");
        mostrarArray(array2);

        System.out.println("Array3");
        mostrarArray(array3);

        //Llamamos al recolector de basura
        System.gc();
    }
    public static void rellenarNumAleatorioArray(int lista[], int a, int b){
        for(int i=0;i<lista.length;i++){
            //Generamos un número entre los parametros pasados
            lista[i]=((int)Math.floor(Math.random()*(a-b)+b));
        }
    }

    public static void mostrarArray(int lista[]){
        for(int i=0;i<lista.length;i++){
            System.out.println("En el indice "+i+" esta el valor "+lista[i]);
        }
    }

    public static int[] multiplicador(int array1[], int array2[]){
        int array3[]=new int[array1.length];
        for(int i=0;i<array1.length;i++){
            array3[i]=array1[i]*array2[i];
        }
        return array3;
    }
}

```

8. Crea un array de números de un tamaño pasado por teclado, el array contendrá números aleatorios entre 1 y 300 y mostrar aquellos números que acaben en un dígito que nosotros le indiquemos por teclado (debes controlar que se introduce un numero correcto), estos deben guardarse en un nuevo array.

Por ejemplo, en un array de 10 posiciones e indicamos mostrar los números acabados en 5, podría salir 155, 25, etc.


```

import javax.swing.JOptionPane;
public class NumTerminadosEnApp {

    public static void main(String[] args) {

        //Indicamos el tamaño
        String texto=JOptionPane.showInputDialog("Introduce un tamaño");
        int num[]=new int[Integer.parseInt(texto)];

        int ultimoDigito;
        do{
            texto=JOptionPane.showInputDialog("Introduce numero entre 0 y 9");
            ultimoDigito=Integer.parseInt(texto);
        }while(!(ultimoDigito>=0 && ultimoDigito<=9));

        //rellenamos el array
        rellenarNumAleatorioArray(num, 1, 300);

        //Creamos un array que contenga los numeros terminados en el numero especificado
        int terminadosEn[]=numTerminadosEn(num, ultimoDigito);

        //Mostramos el resultado, mira el metodo de mostrarArrays
        // mostrarArrayTerminadosEn(num);
        mostrarArrayTerminadosEn(terminadosEn);

    }
    public static void rellenarNumAleatorioArray(int lista[], int a, int b){
        for(int i=0;i<lista.length;i++){
            //Generamos un número entre los parametros pasados
            lista[i]=((int)Math.floor(Math.random()*(a-b)+b));
        }
    }

    public static void mostrarArrayTerminadosEn(int lista[]){
        for(int i=0;i<lista.length;i++){
            //No incluimos las posiciones que tengan un 0
            if(lista[i]!=0){
                System.out.println("El numero "+lista[i]+" acaba en el numero deseado");
            }
        }
    }

    public static int[] numTerminadosEn (int num[], int ultimo_numero){

        //Array que almacenara los numeros terminados en el numero pedido
        int terminadosEn[]=new int[num.length];

        int numeroFinal=0;

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

            /*
             * Restamos el numero por el mismo numero sin unidades
             * Por ejemplo, 325-320=5
             */
            numeroFinal=num[i]-(num[i]/10*10);

            //Si el numero obtenido es el buscado, lo añadimos
            if (numeroFinal==ultimo_numero){
                terminadosEn[i]=num[i];
            }
        }

        return terminadosEn;
    }
}

```

9. Calcula la letra de un DNI, pediremos el DNI por teclado y nos devolverá el DNI completo.

Para calcular la letra, cogeremos el resto de dividir nuestro dni entre 23, el resultado debe estar entre 0 y 22. Haz un método donde según el resultado de la anterior formula busque en un array de caracteres la posición que corresponda a la letra. Esta es la tabla de caracteres:

| POSICION | LETRA |
|----------|-------|
| 0 | T |
| 1 | R |
| 2 | W |
| 3 | A |
| 4 | G |
| 5 | M |
| 6 | Y |
| 7 | F |
| 8 | P |
| 9 | D |
| 10 | X |
| 11 | B |
| 12 | N |
| 13 | J |
| 14 | Z |
| 15 | S |
| 16 | Q |
| 17 | V |
| 18 | H |
| 19 | L |
| 20 | C |
| 21 | K |
| 22 | E |

Por ejemplo, si introduzco 70588387, el resultado sera de 7 que corresponde a 'F'.

```

import javax.swing.JOptionPane;
public class LetraDNIApp {

    public static void main(String[] args) {

        //Declaramos como constante por lo que dividir
        final int DIVISOR=23;

        //Insertamos el DNI
        String texto=JOptionPane.showInputDialog("Escribe los numero de tu DNI");
        int dni=Integer.parseInt(texto);

        //Sacamos el resto
        int res=dni%DIVISOR;

        //Invocamos el metodo
        char letra=letraNIF(res);

        //Mostramos el DNI completo
        System.out.println("Tu DNI es " +dni+letra);

    }
    public static char letraNIF(int res){

        //Definimos el array de char
        char letrasNIF[]={ 'T', 'R', 'W', 'A', 'G', 'M', 'Y',
                           'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z',
                           'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E' };

        //Devolvemos el valor en la posicion del array
        return letrasNIF[res];

    }
}

```

10. Crea un array de números y otro de String de 10 posiciones donde insertaremos notas entre 0 y 10 (debemos controlar que inserte una nota valida), pudiendo ser decimal la nota en el array de números, en el de Strings se insertaran los nombres de los alumnos.

Después, crearemos un array de String donde insertaremos el resultado de la nota con palabras.

- Si la nota esta entre 0 y 4,99 , sera un suspenso
- Si esta entre 5 y 6,99 , sera un bien.
- Si esta entre 7 y 8,99 sera un notable.
- Si esta entre 9 y 10 sera un sobresaliente.

Muestra por pantalla, el alumno su nota y su resultado en palabras. Crea los métodos que creas conveniente.

```

import javax.swing.JOptionPane;
public class NotasApp {

    public static void main(String[] args) {

        //Tamaño del array
        final int TAMANIO=10;

        //Creamos los arrays
        String nombres[]=new String[TAMANIO];
        double notas[]=new double[TAMANIO];

        //Rellenamos el array de una vez
        rellenarArrays(notas, nombres);

        //Devuelve las notas con palabras
        String resultado[]=añadeResultado(notas);

        //Mostramos el resultado
        mostrarArrays(nombres, notas, resultado);

    }

    public static void rellenarArrays(double notas[], String nombres[]){
        for(int i=0;i<notas.length;i++){
            //Introducimos los valores de una vez, se pueden hacer en metodos separados
            nombres[i]=JOptionPane.showInputDialog("Introduce el nombre del alumno "+(i+1));

            double nota;
            //Validamos la nota
            do{
                String texto=JOptionPane.showInputDialog("Introduce una nota del alumno "+(i+1));
                nota=Double.parseDouble(texto);
                System.out.println(nota<=0 || nota>=10);
            }while(nota<=0 || nota>=10);
            notas[i]=nota;
        }
    }

    public static void mostrarArrays(String nombres[], double notas[], String resultado[]){
        for(int i=0;i<nombres.length;i++){
            System.out.println("El alumno "+nombres[i]+" tiene una nota de "+notas[i]+" , por lo que su resultado es "+resultado[i]);
        }
    }

    public static String[] añadeResultado(double notas[]){

        String resultado[]=new String[notas.length];
        for(int i=0;i<notas.length;i++){
            //Convertimos la nota a entero, despues se quedara como estaba
            switch((int)notas[i]){
                case 1:
                case 2:
                case 3:
                case 4:
                    resultado[i]="Suspenso";
                    break;
                case 5:
                case 6:
                    resultado[i]="Bien";
                    break;
                case 7:
                case 8:
                    resultado[i]="Notable";
                    break;
                case 9:
                case 10:
                    resultado[i]="Sobresaliente";
                    break;
            }
        }
        return resultado;
    }
}

```

11. Crea una aplicación que pida un numero por teclado y después comprobaremos si el numero introducido es capicua, es decir, que se lee igual sin importar la dirección.

Por ejemplo, si introducimos 30303 es capicua, si introducimos 30430 no es capicua. Piensa como puedes dar la vuelta al número. Una forma de pasar un número a un array es esta **Character.getNumericValue(cadena.charAt(posicion))**.

```
import java.util.Arrays;
import javax.swing.JOptionPane;
public class CapicuaApp {

    public static void main(String[] args) {

        //Introducimos un numero
        String numero=JOptionPane.showInputDialog("Introduce un número");

        /*
         * Aprovechamos el String para averiguar la longitud del numero,
         * para crear un array compatible, y para dividirlo digitos
         */
        int digitos[]=convierteNumeroAArarray(numero, numero.length());

        //Invocamos el metodo, segun el resultado mostramos un mensaje u otro
        if (EsCapicua(digitos)){
            System.out.println("El numero "+numero+" es capicua");
        }else{
            System.out.println("El numero "+numero+" no es capicua");
        }
    }

    public static int[] convierteNumeroAArarray(String numero, int longitud){

        int digitos[]=new int[longitud];

        for(int i=0;i<digitos.length;i++){
            digitos[i]=Character.getNumericValue(numero.charAt(i));
        }
        return digitos;
    }
}
```

```

public static boolean EsCapicua (int lista[]){

    //Creamos otro array
    int listaprueba[]=new int [lista.length];

    /*
     * Asignamos los valores al nuevo array lo hacemos añadiendo
     * los ultimos valores del primer array, al principio del nuevo array
     * ,es decir, le damos la vuelta al array
     */
    for (int i=0, j=1;j<=lista.length;i++, j++){
        listaprueba[i]=lista[lista.length-j];
    }

    //Usamos el metodo de java.util.Arrays para comparar los arrays
    if (Arrays.equals(lista, listaprueba)){
        return true;
    }

    return false;

}
}

```

12. Dado un array de números de 5 posiciones con los siguiente valores {1,2,3,4,5}, guardar los valores de este array en otro array distinto pero con los valores invertidos, es decir, que el segundo array deba tener los valores {5,4,3,2,1}.

```

public class Ejercicio_DiscoDurodeRoer_Arrays_12 {

    public static void main(String[] args) {

        int[] array1={1,2,3,4,5};

        int[] array2=new int[array1.length];

        for(int i=(array1.length-1),j=0;i>=0;i--,j++){
            array2[j]=array1[i];
            System.out.println(array2[j]);
        }

    }

}

```

13. Rellenar un array de numeros (int) usando el metodo fill de Arrays.


```

import java.util.Arrays;

public class Ejercicio_arrays_DDR_13 {

    public static void main(String[] args) {

        //creo el array
        int num[]=new int[10];

        //relleno todo el array con 20
        Arrays.fill(num, 20);

        //Relleno desde la posicion 3 hasta el final
        Arrays.fill(num, 3, num.length, -1);

        //Muestro el array
        for(int i=0;i<num.length;i++){
            System.out.println(num[i]);
        }

    }

}

```

14. Crear dos arrays, rellenar uno con números y copiarlo al otro usando CopyOf de Arrays.

```

import java.util.Arrays;

public class Ejercicio_arrays_DDR_14 {

    public static void main(String[] args) {

        //Creo el array
        int num[]=new int[10];

        //Rellenamos el array con 1
        Arrays.fill(num, 1);

        //Copiamos de las posiciones 2 a la 9 (7 posiciones tendra)
        int num2[]=Arrays.copyOfRange(num, 2, 9);

        //Muestro el array
        for(int i=0;i<num2.length;i++){
            System.out.println(num2[i]);
        }

    }

}

```

15. Indica si dos arrays creados por ti son iguales con Equals de Arrays.

```

import java.util.Arrays;

public class Ejercicio_arrays_DDR_15 {

    public static void main(String[] args) {

        //Creo el array
        int num1[]=new int[10];
        int num2[]=new int[10];

        //Comparo el contenido de ambos
        if(Arrays.equals(num1, num2)){
            System.out.println("Son iguales");
        }else{
            System.out.println("No son iguales");
        }

    }

}

```

16. Ordena un array de números con Sort de Arrays.

```

import java.util.Arrays;

public class Ejercicio_arrays_DDR_16 {

    public static void main(String[] args) {

        //Creo el array
        int num[]=new int[5];

        //Asigno valores
        num[0]=6;
        num[1]=3;
        num[2]=9;
        num[3]=4;
        num[4]=1;

        //ordeno
        Arrays.sort(num);

        //Muestro el array
        for(int i=0;i<num.length;i++){
            System.out.println(num[i]);
        }

    }

}

```

17. Ordenar un array de números con el método de la burbuja.

```

public class Ejercicio_arrays_DDR_17 {

    public static void main(String[] args) {

        //array a ordenar
        int numeros[]={5,6,2,10,1};

        //llamada funcion
        ordenacionBurbuja(numeros);

        //mostrar el contenido
        for(int i=0;i<numeros.length;i++){
            System.out.println(numeros[i]);
        }

    }

    /**
     * Ordena un array de enteros por el metodo de la burbuja
     * @param lista
     */
    public static void ordenacionBurbuja (int lista[]){
        int cuentaintercambios=0; //Variable que cuenta los intercambios que hacemos
        for (boolean ordenado=false;!ordenado;){ //hasta que no haya mas intercambios no sales
            for (int i=0;i<lista.length-1;i++){ if (lista[i]>lista[i+1]){
                //intercambio
                int variableauxiliar=lista[i];
                lista[i]=lista[i+1];
                lista[i+1]=variableauxiliar;
                //indico que ha habido un intercambio mas
                cuentaintercambios++;
            }
        }
        //Si no hay intercambios, el array esta ordenado
        if (cuentaintercambios==0){
            ordenado=true;
        }
        cuentaintercambios=0; //Reinicio la variable
    }

}

```

18. Ordenar un array de números con el método QuickSort.

```

public class Ejercicio_arrays_DDR_18 {

    public static void main(String[] args) {

        //array a ordenar
        int numeros[]={5,6,2,10,1};

        //llamada funcion
        ordenacionQuicksort(numeros, 0, numeros.length-1);

        //mostramos el contenido
        for(int i=0;i<numeros.length;i++){
            System.out.println(numeros[i]);
        }

    }

    /**
     * Ordena un array por el metodo de ordenacion QuickSort
     * @param array
     * @param izq
     * @param der
     */
    public static void ordenacionQuicksort (int array[], int izq, int der){
        int i=izq;
        int j=der;
        int pivote=array[(i+j)/2]; //determinamos el pivote
        do {
            while (array[i]<pivote){ i++; } while (array[j]>pivote){
                j--;
            }
            if (i<=j){
                //intercambio
                int aux=array[i];
                array[i]=array[j];
                array[j]=aux;
                i++;
                j--;
            }
        }while(i<=j);
        //llamada de funciones (recursivo)
        if (izq<j){
            ordenacionQuicksort(array, izq, j);
        }
        if (i<der){
            ordenacionQuicksort(array, i, der);
        }
    }

}

```

19. Ordenar un array de cadenas con el método Quicksort.

```

public class Ejercicio_arrays_DDR_19 {

    public static void main(String[] args) {

        //array a ordenar
        int numeros[]={5,6,2,10,1};

        //llamada funcion
        ordenacionQuicksort(numeros, 0, numeros.length-1);

        //mostramos el contenido
        for(int i=0;i<numeros.length;i++){
            System.out.println(numeros[i]);
        }

    }

    public static void ordenacionQuicksort (int array[], int izq, int der){
        int i=izq;
        int j=der;
        int pivote=array[(i+j)/2]; //determinamos el pivote
        do {
            while (array[i]<pivote){
                i++;
            }
            while (array[j]>pivote){
                j--;
            }
            if (i<=j){
                //intercambio
                int aux=array[i];
                array[i]=array[j];
                array[j]=aux;
                i++;
                j--;
            }
        }while(i<=j);
        //llamada de funciones (recursivo)
        if (izq<j){
            ordenacionQuicksort(array, izq, j);
        }
        if (i<der){
            ordenacionQuicksort(array, i, der);
        }
    }

}

```

20. Ordenar un array de números con el método que quieras y hacer una búsqueda binaria de cualquier numero.

```

public class Ejercicio_arrays_DDR_20 {

    public static void main(String[] args) {

        //array a ordenar
        int numeros[]={5,6,2,10,1};

        //llamada a la funcion
        ordenacionQuicksort(numeros, 0, numeros.length-1);

        //mostramos el contenido
        for(int i=0;i<numeros.length;i++){
            System.out.println(numeros[i]);
        }

        //realizamos la busqueda binaria
        System.out.println("El numero 5 esta en la posicion "+buscarBin(numeros, 5));

    }

    /**
     * Ordena un array por el metodo de ordenacion QuickSort
     * @param array
     * @param izq
     * @param der
     */
    public static void ordenacionQuicksort (int array[], int izq, int der){
        int i=izq;
        int j=der;
        int pivote=array[(i+j)/2]; //determinamos el pivote
        do {
            while (array[i]<pivote){ i++; } while (array[j]>pivote){
                j--;
            }
            if (i<=j){
                //intercambio
                int aux=array[i];
                array[i]=array[j];
                array[j]=aux;
                i++;
                j--;
            }
        }while(i<=j);
        //llamada de funciones (recursivo)
        if (izq<j){
            ordenacionQuicksort(array, izq, j);
        }
        if (i<der){
            ordenacionQuicksort(array, i, der);
        }
    }
}

```



```

/**
 * Busca un numero entero en un array
 * Necesita que el array este ordenado
 * @param array
 * @param numerobuscado
 * @return posicion del numero
 */
public static int buscarBin (int array[], int numerobuscado){
    int izq=0;
    int der=array.length;
    boolean encontrado=false;
    int pos=-1;

    for(int i=0;!encontrado && i<array.length;i++){
        int central=array[(izq+der)/2]; //elegimos la casilla central
        if (central<=numerobuscado){
            if (central==numerobuscado){ //encontrado
                encontrado=true;
                pos=(izq+der)/2;
            }else{
                izq=(izq+der)/2; //modificamos izq
            }
        }else{
            der=(izq+der)/2; //modificamos der
        }
    }
    return pos;
}
}

```

21. Teniendo un vector con los valores que queramos, meter en otro de la misma longitud, aquellos que sean pares y mayores que 25. Después, mostrar el vector de origen y el de destino, solo los numero introducidos.

```
public class Ejercicio_arrays_DDR_23 {  
  
    public static void main(String[] args) {  
  
        //Creo el array de origen con los numeros que queramos  
        int origen[] = {4, 7, 5, 0, 2, 1, 15, 93, 25, 12, 86, 87, 100};  
  
        //Creamos un array con la misma longitud  
        int destino[] = new int[origen.length];  
  
        //Creamos un contador que nos servira de indice en el array de destino  
        int contador = 0;  
        //Rellenamos el array de destino con la condicion indicada  
        for (int i = 0; i < origen.length; i++) { if (origen[i] % 2 == 0 && origen[i] > 25) {  
            destino[contador] = origen[i]; //usamos contador como indice  
            contador++; //aumentamos el contador  
        }  
    }  
  
    //Muestra el vector de origen  
    System.out.println("Mostrar vector origen");  
    for (int i = 0; i < origen.length; i++) {  
        System.out.println(origen[i]);  
    }  
  
    System.out.println("Mostrar Vector destino");  
    if (contador == 0) { //Si no hay elementos, mostrara un elemento  
        System.out.println("No hay numeros que cumplan la condicion");  
    } else {  
        //Mostrar el vector de origen  
        for (int i = 0; i < contador; i++) {  
            System.out.println(destino[i]);  
        }  
    }  
}  
}
```