Contenido

[**SECCION 12 ARREGLOS** 2](#_Toc139547227)

[93. Introducción a los arreglos 2](#_Toc139547228)

[94. Ejemplo de un arreglo de enteros 2](#_Toc139547229)

[95. Ejemplo de un arreglo de string y ordenar con sort 3](#_Toc139547230)

[96. Iterando arreglos 4](#_Toc139547231)

[97. Iterando en orden inverso un arreglo 6](#_Toc139547232)

[98. Modificando el arreglo en reverso 8](#_Toc139547233)

[Practica 1 10](#_Toc139547234)

[Practica 2 11](#_Toc139547235)

[Practica 3 12](#_Toc139547236)

[99. Ordenamiento algoritmo burbuja 13](#_Toc139547237)

[100. Ordenamiento algoritmo burbuja método optimizado 14](#_Toc139547238)

[101. Reutilizando algoritmo burbuja 14](#_Toc139547239)

[102. Imprimiendo el i-ésimo número por el principio y el i-ésimo por el final 14](#_Toc139547240)

[103. Combinar dos arreglos en un tercero 14](#_Toc139547241)

[104. Combinar dos arreglos en un tercero con múltiples elementos 14](#_Toc139547242)

[105. Obtener el número mayor de un arreglo de enteros 14](#_Toc139547243)

[106. Detectar orden del arreglo 14](#_Toc139547244)

[107. Copiar pares e impares de un arreglo de enteros y en sus respectivos arreglos 14](#_Toc139547245)

[108. Sistema de notas de alumnos por asignaturas 14](#_Toc139547246)

[109. Buscar elementos en un arreglo 14](#_Toc139547247)

[110. Mover los elementos en una posición desde el inicio del arreglo hasta el final 14](#_Toc139547248)

[111. Agregar un elemento en una posición indicada y mover los siguientes elementos 14](#_Toc139547249)

[112. Agregar un elemento en una posición adecuada de manera que se mantenga… 14](#_Toc139547250)

[113. Eliminar un elemento del arreglo 14](#_Toc139547251)

[114. Insertando un nuevo elemento e incrementando el tamaño del arreglo 14](#_Toc139547252)

[115. Descargar Código Fuente 14](#_Toc139547253)

[Tarea 13: Ordenar el arreglo de la forma último primero, etc 14](#_Toc139547254)

[Tarea 14: Imprima el numero mas alto del arreglo 15](#_Toc139547255)

[Tarea 15: Sistema estadístico de un arreglo 15](#_Toc139547256)

[Tarea 16: Mayor ocurrencia en el arreglo 15](#_Toc139547257)

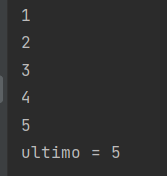
[Tarea 17: Imprimir Histograma 15](#_Toc139547258)

# **SECCION 12 ARREGLOS**

# 93. Introducción a los arreglos

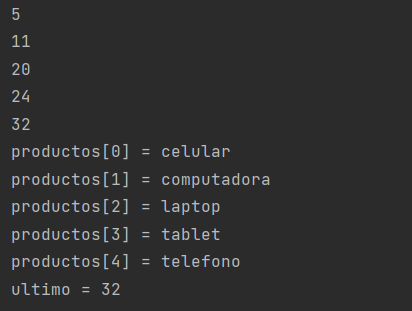
# 94. Ejemplo de un arreglo de enteros

package \_94\_EjemploArregloEnteros;  
  
public class EjemploArreglos {  
  
 public static void main(String[] args) {  
  
 // Declaración e inicialización de un arreglo de enteros con capacidad para 5 elementos  
 int [] enteros = new int[5];  
  
 // Asignación del valor elemento del arreglo  
 enteros[0] = 1;  
 enteros[1] = Integer.*valueOf*("2"); // Convierte la cadena "2" en un valor entero y lo asigna al segundo elemento del arreglo  
 enteros[2] = 3;  
 enteros[3] = 4;  
 enteros[4] = 5;  
  
 // Imprime elemento del arreglo  
 System.*out*.println(enteros[0]);  
 System.*out*.println(enteros[1]);  
 System.*out*.println(enteros[2]);  
 System.*out*.println(enteros[3]);  
 System.*out*.println(enteros[4]);  
  
 //Obtener ultimo eemento del arreglo  
 int ultimo = enteros[enteros.length-1];  
  
 System.*out*.println("ultimo = " + ultimo);  
  
  
 }  
}



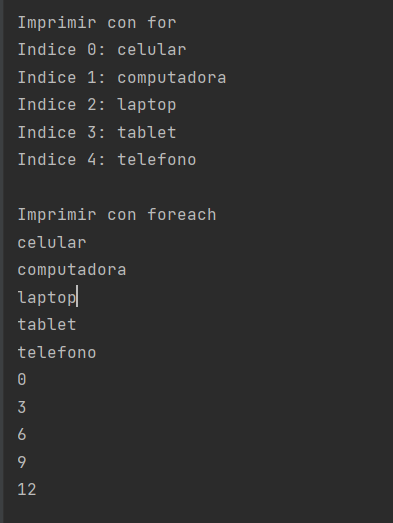
# 95. Ejemplo de un arreglo de string y ordenar con sort

package \_95\_EjemploArregloStringAndOrdenarWithSort;  
  
import java.util.Arrays;  
  
public class EjemploArreglos {  
  
 public static void main(String[] args) {  
  
 // Declaración e inicialización de un arreglo de enteros con capacidad para 5 elementos  
 int [] enteros = new int[5];  
 String [] productos = new String[5];  
  
 // Asignación del valor elemento del arreglo  
 enteros[0] = 11;  
 enteros[1] = Integer.*valueOf*("20"); // Convierte la cadena "2" en un valor entero y lo asigna al segundo elemento del arreglo  
 enteros[2] = 32;  
 enteros[3] = 24;  
 enteros[4] = 5;  
  
 productos[0] = "telefono";  
 productos[1] = "tablet";  
 productos[2] = "laptop";  
 productos[3] = "computadora";  
 productos[4] = "celular";  
  
 //ORDENA NUMEROS  
 Arrays.*sort*(enteros);// Ordena los elementos del arreglo de enteros en orden ascendente  
  
 // Imprime elemento del arreglo  
 System.*out*.println(enteros[0]);  
 System.*out*.println(enteros[1]);  
 System.*out*.println(enteros[2]);  
 System.*out*.println(enteros[3]);  
 System.*out*.println(enteros[4]);  
  
 //ORDENA STRING  
 Arrays.*sort*(productos);//// Ordena los elementos del arreglo de productos en orden alfabético  
  
 System.*out*.println("productos[0] = " + productos[0]);  
 System.*out*.println("productos[1] = " + productos[1]);  
 System.*out*.println("productos[2] = " + productos[2]);  
 System.*out*.println("productos[3] = " + productos[3]);  
 System.*out*.println("productos[4] = " + productos[4]);  
  
 //Obtener ultimo elmento del arreglo (NUMEROS)  
 int ultimo = enteros[enteros.length-1];  
 System.*out*.println("ultimo = " + ultimo);  
  
  
 }  
}



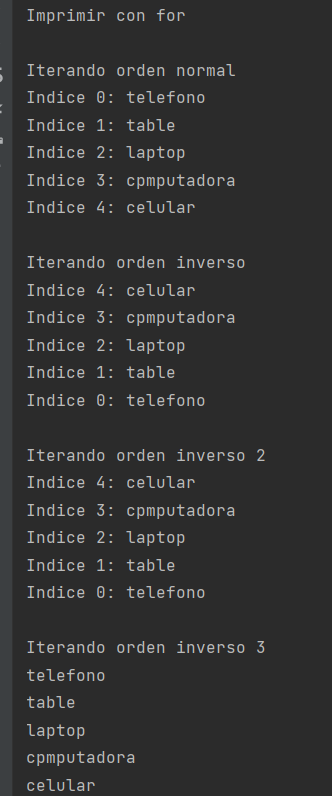
# 96. Iterando arreglos

package \_96\_IterandoArreglo;  
  
import java.util.Arrays;  
  
public class EjemploArreglosFor {  
  
 public static void main(String[] args) {  
  
 // Declaración e inicialización de un arreglo de enteros con capacidad para 5 elementos  
 int [] enteros = new int[5];  
 String[] productos = new String[5];  
  
 // Asignación del valor elemento del arreglo  
 enteros[0] = 11;  
 enteros[1] = Integer.*valueOf*("20"); // Convierte la cadena "2" en un valor entero y lo asigna al segundo elemento del arreglo  
 enteros[2] = 32;  
 enteros[3] = 24;  
 enteros[4] = 5;  
  
 productos[0] = "telefono";  
 productos[1] = "tablet";  
 productos[2] = "laptop";  
 productos[3] = "computadora";  
 productos[4] = "celular";  
  
  
 Arrays.*sort*(productos);  
  
 System.*out*.println("\nImprimir con for");  
 int totalString = productos.length; // Obtiene la longitud del arreglo productos  
 for (int i = 0; i < totalString; i++) { // Itera sobre los elementos del arreglo utilizando un bucle for  
 System.*out*.println("Indice " + i + ": " + productos[i]); // Imprime el índice y el valor del elemento actual del arreglo  
 }  
  
 System.*out*.println("\nImprimir con foreach");  
 for (String p : productos) { // Itera sobre los elementos del arreglo utilizando un bucle for-each  
 System.*out*.println(p); // Imprime el valor del elemento actual del arreglo  
 }  
  
 //IMPRIMER NUMEROS ENTEROS  
 int totalEnteros = enteros.length;  
 for (int i = 0; i < enteros.length; i++) {  
 enteros[i] = i \* 3; // Asignación del valor al elemento del arreglo multiplicado por 3  
 System.*out*.println(enteros[i]); // Imprime el valor del elemento actual del arreglo  
 }  
  
 }  
}



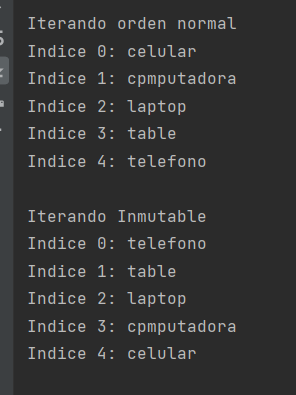
# 97. Iterando en orden inverso un arreglo

package \_97\_IteraArregloOdenDescendente;  
  
import java.util.Arrays;  
import java.util.Collections;  
  
public class EjemploArreglosForDescendente {  
  
 public static void main(String[] args) {  
  
 // Declaración e inicialización de un arreglo de enteros con capacidad para 5 elementos  
 //int [] enteros = new int[5];  
 //String[] productos = new String[5];  
  
// // Asignación del valor elemento del arreglo  
// enteros[0] = 11;  
// enteros[1] = Integer.valueOf("20"); // Convierte la cadena "2" en un valor entero y lo asigna al segundo elemento del arreglo  
// enteros[2] = 32;  
// enteros[3] = 24;  
// enteros[4] = 5;  
  
// productos[0] = "telefono";  
// productos[1] = "tablet";  
// productos[2] = "laptop";  
// productos[3] = "computadora";  
// productos[4] = "celular";  
  
  
 String[] productos = {"telefono","table","laptop","cpmputadora","celular"};  
  
 Arrays.*sort*(productos);  
  
 System.*out*.println("\nImprimir con for");  
 System.*out*.println("\nIterando orden normal");  
 int totalString = productos.length; // Obtiene la longitud del arreglo productos  
 for (int i = 0; i < totalString; i++) { // Itera sobre los elementos del arreglo utilizando un bucle for  
 System.*out*.println("Indice " + i + ": " + productos[i]); // Imprime el índice y el valor del elemento actual del arreglo  
 }  
  
 System.*out*.println("\nIterando orden inverso");  
 for (int i = 0; i < totalString; i++) {  
 System.*out*.println("Indice " + (totalString-1-i) + ": " + productos[totalString-1-i]);  
 }  
  
 System.*out*.println("\nIterando orden inverso 2");  
 for (int i = totalString-1; i >= 0; i--) {  
 System.*out*.println("Indice " + (i) + ": " + productos[i]);  
 }  
  
 System.*out*.println("\nIterando orden inverso 3");  
 // Utiliza Arrays.sort() con un Comparator para ordenar en orden descendente  
 Arrays.*sort*(productos, Collections.*reverseOrder*());  
  
 // Imprime los elementos del arreglo en orden descendente  
 for (String producto : productos) {  
 System.*out*.println(producto);  
 }  
  
 }  
}



# 98. Modificando el arreglo en reverso

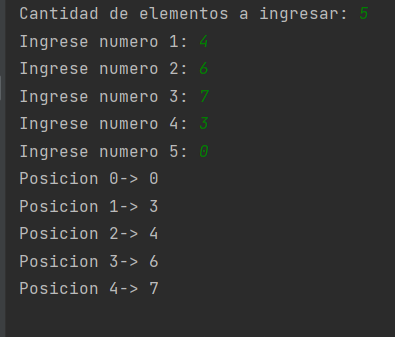
package \_98\_ModificandoArregloReverse;  
  
import java.util.Arrays;  
import java.util.Collections;  
  
public class EjemploArreglosInversoMutable {  
  
 public static void main(String[] args) {  
  
 String[] productos = {"telefono","table","laptop","cpmputadora","celular"};  
  
 Arrays.*sort*(productos);  
  
 System.*out*.println("\nIterando orden normal");  
 int totalString = productos.length; // Obtiene la longitud del arreglo productos  
 for (int i = 0; i < totalString; i++) { // Itera sobre los elementos del arreglo utilizando un bucle for  
 System.*out*.println("Indice " + i + ": " + productos[i]); // Imprime el índice y el valor del elemento actual del arreglo  
 }  
  
 System.*out*.println("\nIterando Inmutable");  
// for (int i = 0; i < totalString/2; i++) {  
// String actual = productos[i]; // Almacena el valor actual en una variable temporal  
// String inverso = productos[totalString-1-i]; // Almacena el valor inverso en otra variable temporal  
// productos[i] = inverso; // Asigna el valor inverso al elemento actual  
// productos[totalString-1-i] = actual; // Asigna el valor actual al elemento inverso  
// }  
  
 //LO MISMO DE ARRIBA COMENTADO PERO USANDO METODOS  
 Collections.*reverse*(Arrays.*asList*(productos)); // Invierte el orden de los elementos en el arreglo "productos"  
  
  
 for (int i = 0; i < totalString; i++) { // Itera sobre los elementos del arreglo utilizando un bucle for  
 System.*out*.println("Indice " + i + ": " + productos[i]); // Imprime el índice y el valor del elemento actual del arreglo  
 }  
  
 }  
}



# Practica 1

Escribe un programa que solicite al usuario ingresar una serie de números enteros.  
El programa debe almacenar estos números en un arreglo de tipo int.  
Luego, el programa debe ordenar el arreglo de forma ascendente y mostrar los números ordenados.

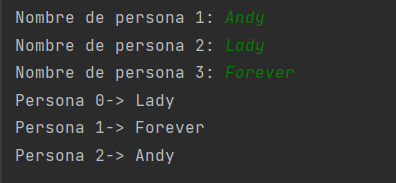
package Practicas.Practica1;  
  
import java.util.Arrays;  
import java.util.Scanner;  
  
public class ArreglosEnterosInOrderAscendente {  
 public static void main(String[] args) {  
  
 Scanner entrada = new Scanner(System.*in*);  
  
 System.*out*.print("Cantidad de elementos a ingresar: ");  
 int cantidad = entrada.nextInt();  
  
 int numeros[] = new int[cantidad];  
  
  
 //INGRESANDO DATOS  
 for (int i = 0; i<cantidad; i++){  
 System.*out*.print("Ingrese numero "+(i+1)+": ");  
 numeros[i] = entrada.nextInt();  
 entrada.nextLine();  
 }  
  
 Arrays.*sort*(numeros);  
  
 //MOSTRANDO DATOS  
 for (int i = 0; i<cantidad; i++){  
 System.*out*.println("Posicion "+(i)+"-> "+numeros[i]);  
 }  
  
  
 }  
}



# Practica 2

Escribe un programa que solicite al usuario ingresar una lista de nombres de personas.  
El programa debe almacenar estos nombres en un arreglo de tipo String.  
Luego, el programa debe ordenar el arreglo en orden alfabético descendente y mostrar los nombres en ese nuevo orden.

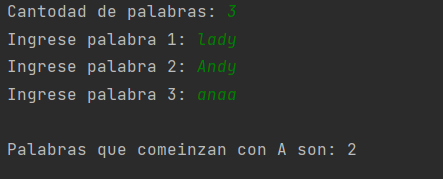
package Practicas.Practica2;  
  
import java.util.Arrays;  
import java.util.Collections;  
import java.util.Scanner;  
  
public class ListaPersonasOrdenDescendnete {  
 public static void main(String[] args) {  
  
 Scanner entrada = new Scanner(System.*in*);  
  
 System.*out*.print("Cantidad de personas a ingresar: ");  
 int cantidad = entrada.nextInt();  
  
 String personas[] = new String[cantidad];  
  
 for (int i = 0; i<cantidad; i++){  
 System.*out*.print("Nombre de persona "+(i+1)+": ");  
 personas[i] = entrada.next();  
 //entrada.nextLine();  
 }  
  
  
 Arrays.*sort*(personas, Collections.*reverseOrder*());  
  
 for (int i = 0; i<cantidad; i++){  
 System.*out*.println("Persona "+(i)+"-> "+personas[i]);  
 }  
  
  
 }  
}



# Practica 3

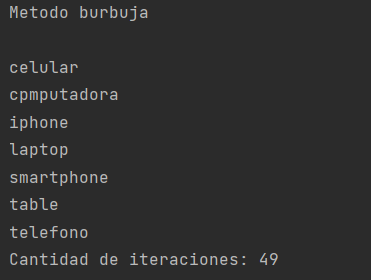
Escribe un programa que solicite al usuario ingresar una serie de palabras.  
El programa debe almacenar estas palabras en un arreglo de tipo String.  
Luego, el programa debe contar la cantidad de palabras que comienzan con la letra "A" y mostrar dicho conteo.

package Practicas.Practica3;  
  
import java.util.Scanner;  
  
public class ComienzanConLetra {  
 public static void main(String[] args) {  
  
 Scanner entrada = new Scanner(System.*in*);  
  
 int contador = 0;  
  
 System.*out*.print("Cantodad de palabras: ");  
 int cantidad = entrada.nextInt();  
  
 String palabras[] = new String[cantidad];  
  
 for (int i = 0; i<cantidad; i++){  
 System.*out*.print("Ingrese palabra "+(i+1)+ ": ");  
 palabras[i] = entrada.next();  
  
 // Obtener la primera letra de la palabra  
 char primeraLetra = palabras[i].charAt(0);  
  
 // Verificar si la primera letra es 'A' o 'a'  
 if (primeraLetra == 'A' || primeraLetra == 'a') {  
 contador++;  
 }  
  
 }  
  
 System.*out*.println("\nPalabras que comeinzan con A son: " +contador );  
  
 }  
}



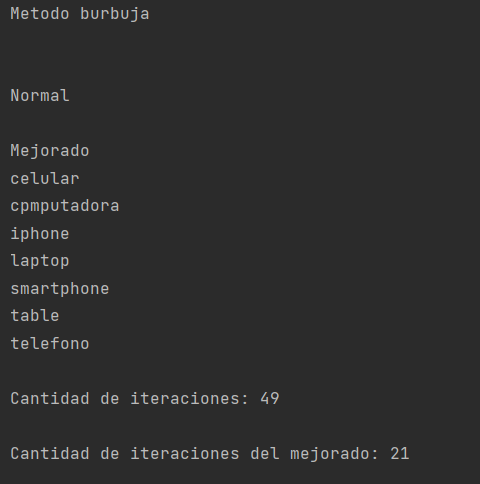
# 99. Ordenamiento algoritmo burbuja

package \_99\_OrdenamientoBurbuja;  
  
public class Burbuja {  
 public static void main(String[] args) {  
  
 String[] productos = {"telefono", "table", "laptop", "cpmputadora", "celular","smartphone","iphone"};  
  
 System.*out*.println("Metodo burbuja\n");  
  
 int total = productos.length;  
 int contador = 0;  
  
 for (int i = 0; i < total; i++) {  
 for (int j = 0; j < total; j++) {  
 if (productos[i].compareTo(productos[j]) < 0) {  
 String auxiliar = productos[i];  
 productos[i] = productos[j];  
 productos[j] = auxiliar;  
 }  
 contador++;  
 }  
 }  
  
 for (String p : productos){  
 System.*out*.println(p);  
 }  
  
 System.*out*.println("Cantidad de iteraciones: "+contador);  
  
 }  
}



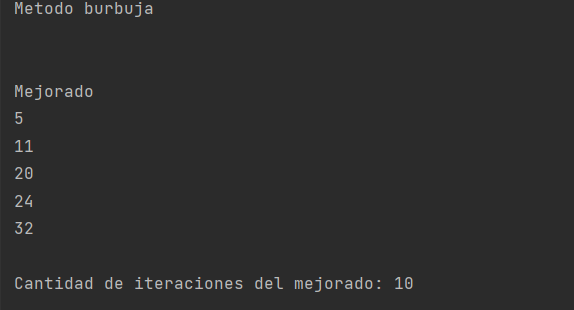
# 100. Ordenamiento algoritmo burbuja método optimizado

package \_100\_BurbujaMetodoOptimizado;  
  
public class BurbujaOptimizado {  
 public static void main(String[] args) {  
  
 String[] productos = {"telefono", "table", "laptop", "cpmputadora", "celular", "smartphone", "iphone"};  
  
 System.*out*.println("Metodo burbuja\n");  
  
 int total = productos.length;  
 int contador = 0;  
 int contadorDelMejorado = 0;  
  
 System.*out*.println("\nNormal");  
  
 for (int i = 0; i < total; i++) {  
 for (int j = 0; j < total; j++) {  
 // Compara el elemento en la posición "i" con el elemento en la posición "j"  
 // utilizando el método compareTo() de la clase String.  
 // Si el resultado de la comparación es menor que 0, significa que el elemento en la posición "i" es menor que el elemento en la posición "j",  
 // por lo tanto, se debe realizar un intercambio.  
 if (productos[i].compareTo(productos[j]) < 0) {  
 // Realiza el intercambio de elementos utilizando una variable auxiliar.  
 // Asigna el valor del elemento en la posición "i" a la variable auxiliar.  
 String auxiliar = productos[i];  
 // Asigna el valor del elemento en la posición "j" al elemento en la posición "i".  
 productos[i] = productos[j];  
 // Asigna el valor de la variable auxiliar al elemento en la posición "j".  
 productos[j] = auxiliar;  
 }  
 // Incrementa un contador para llevar un seguimiento de las comparaciones realizadas.  
 contador++;  
 }  
 }  
  
  
 System.*out*.println("\nMejorado");  
  
 for (int i = 0; i < total - 1; i++) {  
 for (int j = 0; j < total - 1 - i; j++) {  
 // Compara el elemento en la posición "j+1" con el elemento en la posición "j"  
 // utilizando el método compareTo() de la clase String.  
 // Si el resultado de la comparación es menor que 0, significa que el elemento en la posición "j+1" es menor que el elemento en la posición "j",  
 // por lo tanto, se debe realizar un intercambio.  
 if (productos[j + 1].compareTo(productos[j]) < 0) {  
 // Realiza el intercambio de elementos utilizando una variable auxiliar.  
 // Asigna el valor del elemento en la posición "j" a la variable auxiliar.  
 String auxiliar = productos[j];  
 // Asigna el valor del elemento en la posición "j+1" al elemento en la posición "j".  
 productos[j] = productos[j + 1];  
 // Asigna el valor de la variable auxiliar al elemento en la posición "j+1".  
 productos[j + 1] = auxiliar;  
 }  
 // Incrementa un contador para llevar un seguimiento de las comparaciones realizadas.  
 contadorDelMejorado++;  
 }  
 }  
  
  
 for (String p : productos) {  
 System.*out*.println(p);  
 }  
  
 System.*out*.println("\nCantidad de iteraciones: " + contador);  
 System.*out*.println("\nCantidad de iteraciones del mejorado: " + contadorDelMejorado);  
  
 }  
}



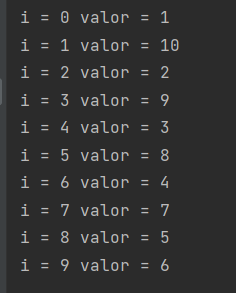
# 101. Reutilizando algoritmo burbuja

package \_101\_ReutilizandoMetodoBurbujaEnteros;  
  
public class BurbujaOptimizadoParaEnteros {  
 public static void main(String[] args) {  
  
  
 System.*out*.println("Metodo burbuja\n");  
  
 int[] enteros = new int[5];  
  
 // Asignación del valor elemento del arreglo  
 enteros[0] = 11;  
 enteros[1] = Integer.*valueOf*("20"); // Convierte la cadena "2" en un valor entero y lo asigna al segundo elemento del arreglo  
 enteros[2] = 32;  
 enteros[3] = 24;  
 enteros[4] = 5;  
  
 int total = enteros.length;  
// int contador = 0;  
 int contadorDelMejorado = 0;  
  
  
 System.*out*.println("\nMejorado");  
  
 for (int i = 0; i < enteros.length - 1; i++) {  
 for (int j = 0; j < enteros.length - 1 - i; j++) {  
 // Dado que los elementos son de tipo int, no se puede utilizar el método compareTo().  
 // En su lugar, se realiza un cast del elemento en la posición "j+1" a Integer y se utiliza el método compareTo() de Integer.  
 if (((Integer) enteros[j + 1]).compareTo(enteros[j]) < 0) {//al sert int no puede usar compare to, pero puedo hacer un cast para no cambiar int a Integer  
 int auxiliar = enteros[j];  
 enteros[j] = enteros[j + 1];  
 enteros[j + 1] = auxiliar;  
 }  
 contadorDelMejorado++;  
 }  
 }  
  
 for (int p : enteros) {  
 System.*out*.println(p);  
 }  
  
 //System.out.println("\nCantidad de iteraciones: " + contador);  
 System.*out*.println("\nCantidad de iteraciones del mejorado: " + contadorDelMejorado);  
  
 }  
}



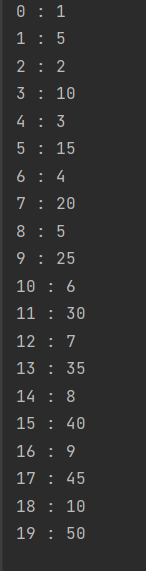
# 102. Imprimiendo el i-ésimo número por el principio y el i-ésimo por el final

package \_102\_IesimoNumero;  
  
public class EjemploArregloOrdenPrincipioFinal {  
 public static void main(String[] args) {  
  
 int numeros[] = new int[10]; // Crear un arreglo llamado "numeros" con capacidad para 10 elementos  
 int a[] = new int[10]; // Crear un arreglo llamado "a" con capacidad para 10 elementos  
  
 // Llenar el arreglo "numeros" con valores del 1 al 10  
 for (int i = 0; i < numeros.length; i++) {  
 numeros[i] = i + 1;  
 }  
  
 int aux = 0;  
 // Recorrer el arreglo "numeros" de forma iterativa  
 // y copiar sus elementos en el arreglo "a" de manera alternada  
 for (int i = 0; i < numeros.length - i; i++) {  
 a[aux++] = numeros[i]; // Copiar el i-ésimo elemento de "numeros" en "a"  
 a[aux++] = numeros[numeros.length - 1 - i]; // Copiar el elemento opuesto de "numeros" en "a"  
 }  
  
 // Imprimir el contenido del arreglo "a" en la consola  
 for (int i = 0; i < a.length; i++) {  
 System.*out*.println("i = " + i + " valor = " + a[i]);  
 }  
 }  
}



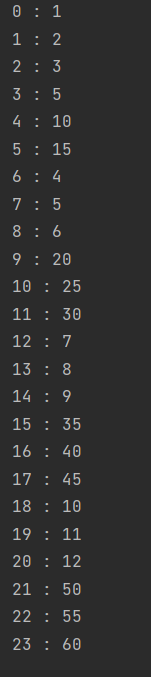
# 103. Combinar dos arreglos en un tercero

package \_103\_Combina2Arreglos;  
  
public class CombinaArreglos {  
 public static void main(String[] args) {  
  
 int [] a, b, c; // Declarar tres arreglos: "a", "b" y "c"  
  
 a = new int[10]; // Crear el arreglo "a" con capacidad para 10 elementos  
 b = new int[10]; // Crear el arreglo "b" con capacidad para 10 elementos  
 c = new int[20]; // Crear el arreglo "c" con capacidad para 20 elementos  
  
 // Llenar el arreglo "a" con valores del 1 al 10  
 for (int i = 0; i < a.length; i++) {  
 a[i] = i + 1;  
 }  
  
 // Llenar el arreglo "b" con valores del 5 al 50, incrementando de 5 en 5  
 for (int i = 0; i < a.length; i++) {  
 b[i] = (i + 1) \* 5;  
 }  
  
 int aux = 0;  
 // Combinar los arreglos "a" y "b" en el arreglo "c" de forma alternada  
 for (int i = 0; i < 10; i++) {  
 c[aux++] = a[i]; // Copiar el i-ésimo elemento de "a" en "c"  
 c[aux++] = b[i]; // Copiar el i-ésimo elemento de "b" en "c"  
 }  
  
 // Imprimir el contenido del arreglo "c" en la consola  
 for (int i = 0; i < c.length; i++) {  
 System.*out*.println(i + " : " + c[i]);  
 }  
 }  
}



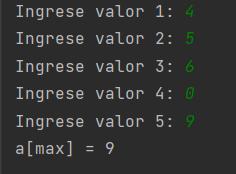
# 104. Combinar dos arreglos en un tercero con múltiples elementos

package \_104\_Combina2ArrreglosEnUnTerceroConMultiplesElementos;  
  
public class CombinaArreglos {  
 public static void main(String[] args) {  
  
 int[] a, b, c;  
  
 a = new int[12]; // Crear el arreglo "a" con capacidad para 12 elementos  
 b = new int[12]; // Crear el arreglo "b" con capacidad para 12 elementos  
 c = new int[24]; // Crear el arreglo "c" con capacidad para 24 elementos  
  
 // Llenar el arreglo "a" con valores del 1 al 12  
 for (int i = 0; i < a.length; i++) {  
 a[i] = i + 1;  
 }  
  
 // Llenar el arreglo "b" con valores del 5 al 60, incrementando de 5 en 5  
 for (int i = 0; i < a.length; i++) {  
 b[i] = (i + 1) \* 5;  
 }  
  
 int aux = 0;  
 // Combinar los arreglos "a" y "b" en el arreglo "c" en bloques de 3 elementos  
 for (int i = 0; i < b.length; i += 3) {  
 // Copiar 3 elementos de "a" al arreglo "c"  
 for (int j = 0; j < 3; j++) {  
 c[aux++] = a[i + j];  
 }  
 // Copiar 3 elementos de "b" al arreglo "c"  
 for (int j = 0; j < 3; j++) {  
 c[aux++] = b[i + j];  
 }  
 }  
  
 // Imprimir el contenido del arreglo "c" en la consola  
 for (int i = 0; i < c.length; i++) {  
 System.*out*.println(i + " : " + c[i]);  
 }  
 }  
}



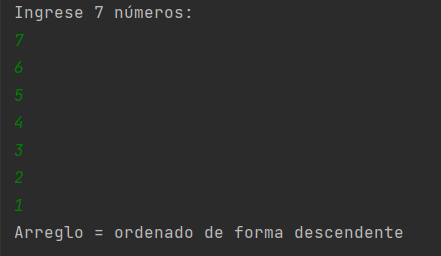
# 105. Obtener el número mayor de un arreglo de enteros

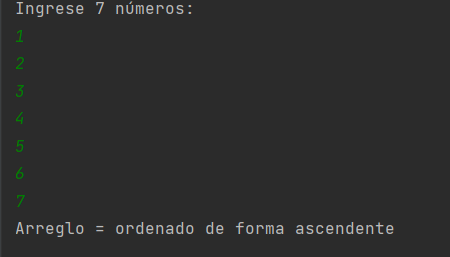
package \_105\_MayorElementoAreeglo;  
  
import java.util.Scanner;  
  
public class ElementoMayorArreglo {  
 public static void main(String[] args) {  
  
 Scanner entrada = new Scanner(System.*in*);  
  
 int a[] = new int[5]; // Crear el arreglo "a" con capacidad para 5 elementos  
  
 // Solicitar al usuario que ingrese los valores del arreglo "a"  
 for (int i = 0; i < a.length; i++) {  
 System.*out*.print("Ingrese valor " + (i + 1) + ": ");  
 a[i] = entrada.nextInt();  
 }  
  
 int max = 0;  
 // Encontrar el índice del elemento máximo en el arreglo "a"  
 for (int i = 1; i < a.length; i++) {  
 max = (a[max] > a[i]) ? max : i;  
 }  
  
 //LO MISMO DE ARRIBA PERO EN FORMATO IF ELSE   
// int max = 0;  
// // Encontrar el índice del elemento máximo en el arreglo "a"  
// for (int i = 1; i < a.length; i++) {  
// if (a[max] > a[i]) {  
// max = max;  
// } else {  
// max = i;  
// }  
// }  
  
 System.*out*.println("a[max] = " + a[max]); // Imprimir el valor del elemento máximo en el arreglo "a"  
  
 }  
}

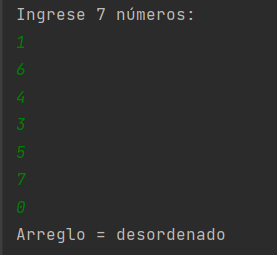


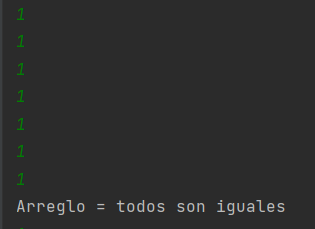
# 106. Detectar orden del arreglo

package \_106\_DetectarodenArreglo;  
import java.util.Scanner;  
  
public class EjemploArreglosDetectarOrden {  
 public static void main(String[] args) {  
  
 int[] a = new int[7]; // Crear el arreglo "a" con capacidad para 7 elementos  
  
 Scanner s = new Scanner(System.*in*);  
 System.*out*.println("Ingrese 7 números:");  
 for(int i = 0; i < a.length; i++){  
 a[i] = s.nextInt(); // Solicitar al usuario que ingrese los números y almacenarlos en el arreglo "a"  
 }  
  
 boolean ascendente = false; // Bandera para indicar si el arreglo está ordenado de forma ascendente  
 boolean descendente = false; // Bandera para indicar si el arreglo está ordenado de forma descendente  
  
 // Verificar el orden de los elementos en el arreglo  
 for(int i = 0; i < a.length - 1; i++){  
 if(a[i] > a[i+1]){  
 descendente = true; // Existe un elemento mayor seguido de un elemento menor, indicando un orden descendente  
 }  
 if(a[i] < a[i+1]){  
 ascendente = true; // Existe un elemento menor seguido de un elemento mayor, indicando un orden ascendente  
 }  
 }  
  
 // Determinar el tipo de orden del arreglo y mostrar el resultado  
 if(ascendente == true && descendente == true){  
 System.*out*.println("Arreglo = desordenado");  
 }  
 if(ascendente == false && descendente == false){  
 System.*out*.println("Arreglo = todos son iguales");  
 }  
 if(ascendente == true && descendente == false){  
 System.*out*.println("Arreglo = ordenado de forma ascendente");  
 }  
 if(ascendente == false && descendente == true){  
 System.*out*.println("Arreglo = ordenado de forma descendente");  
 }  
 }  
}



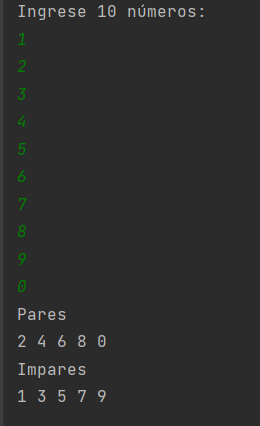






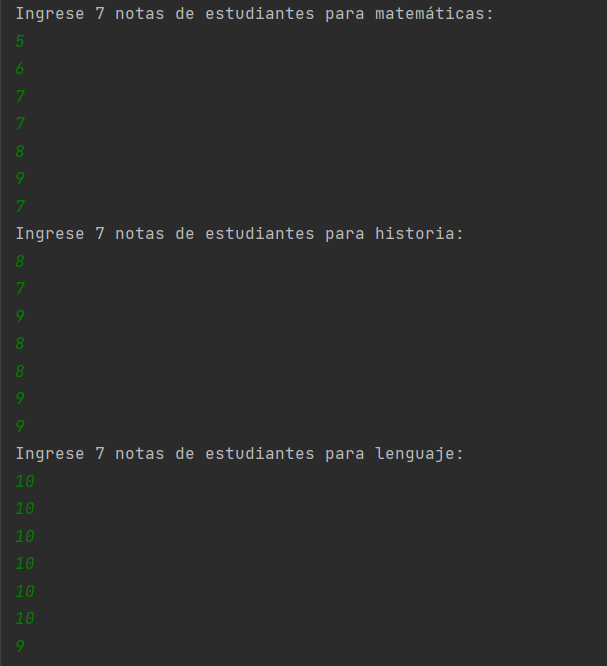
# 107. Copiar pares e impares de un arreglo de enteros y en sus respectivos arreglos

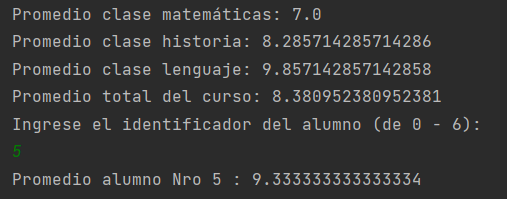
package \_107\_CopiaParImparEnSuPopioArreglo;  
  
import java.util.Scanner;  
  
public class EjemploArreglosParesImpares {  
 public static void main(String[] args) {  
  
 int[] a, pares, impares;  
 int totalPares = 0, totalImpares = 0;  
 a = new int[10]; // Crear el arreglo "a" con capacidad para 10 elementos  
 Scanner s = new Scanner(System.*in*);  
  
 System.*out*.println("Ingrese 10 números:");  
 for(int i = 0; i < a.length; i++){  
 a[i] = s.nextInt(); // Solicitar al usuario que ingrese los números y almacenarlos en el arreglo "a"  
 }  
  
 // Contar la cantidad de números pares e impares en el arreglo "a"  
 for(int i = 0; i < a.length; i++){  
 if(a[i] % 2 == 0){  
 totalPares++; // Incrementar el contador de números pares  
 } else {  
 totalImpares++; // Incrementar el contador de números impares  
 }  
 }  
  
 pares = new int[totalPares]; // Crear el arreglo "pares" con capacidad para almacenar los números pares  
 impares = new int[totalImpares]; // Crear el arreglo "impares" con capacidad para almacenar los números impares  
  
 int j = 0;  
 int k = 0;  
 // Copiar los números pares e impares del arreglo "a" a los arreglos "pares" e "impares", respectivamente  
 for(int i = 0; i < a.length; i++){  
 if(a[i] % 2 == 0){  
 pares[j++] = a[i]; // Copiar el número par a "pares"  
 } else {  
 impares[k++] = a[i]; // Copiar el número impar a "impares"  
 }  
 }  
  
 System.*out*.println("Pares");  
 // Imprimir los números pares  
 for(int i = 0; i < pares.length; i++){  
 System.*out*.print(pares[i] + " ");  
 }  
  
 System.*out*.println("\r\nImpares");  
 // Imprimir los números impares  
 for(int i = 0; i < impares.length; i++){  
 System.*out*.print(impares[i] + " ");  
 }  
 System.*out*.println();  
 }  
}



# 108. Sistema de notas de alumnos por asignaturas

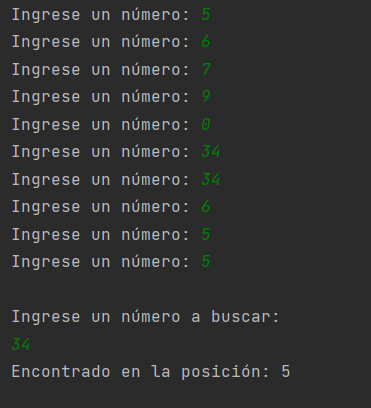
import java.util.Scanner;  
  
public class EjemploArregloNotasAlumnos {  
 public static void main(String[] args) {  
  
 double[] claseMatematicas, claseHistoria, claseLenguaje;  
 double sumNotasMatematicas = 0, sumNotasHistoria = 0, sumNotasLenguaje = 0;  
 claseMatematicas = new double[7]; // Crear el arreglo "claseMatematicas" para almacenar las notas de matemáticas  
 claseHistoria = new double[7]; // Crear el arreglo "claseHistoria" para almacenar las notas de historia  
 claseLenguaje = new double[7]; // Crear el arreglo "claseLenguaje" para almacenar las notas de lenguaje  
  
 Scanner s = new Scanner(System.*in*);  
 System.*out*.println("Ingrese 7 notas de estudiantes para matemáticas:");  
 for(int i = 0; i < claseMatematicas.length; i++){  
 claseMatematicas[i] = s.nextDouble(); // Solicitar al usuario que ingrese las notas de matemáticas y almacenarlas en el arreglo "claseMatematicas"  
 }  
  
 System.*out*.println("Ingrese 7 notas de estudiantes para historia:");  
 for(int i = 0; i < claseHistoria.length; i++){  
 claseHistoria[i] = s.nextDouble(); // Solicitar al usuario que ingrese las notas de historia y almacenarlas en el arreglo "claseHistoria"  
 }  
  
 System.*out*.println("Ingrese 7 notas de estudiantes para lenguaje:");  
 for(int i = 0; i < claseLenguaje.length; i++){  
 claseLenguaje[i] = s.nextDouble(); // Solicitar al usuario que ingrese las notas de lenguaje y almacenarlas en el arreglo "claseLenguaje"  
 }  
  
 // Calcular la suma de las notas de cada materia  
 for(int i = 0; i < 7; i++){  
 sumNotasMatematicas += claseMatematicas[i];  
 sumNotasHistoria += claseHistoria[i];  
 sumNotasLenguaje += claseLenguaje[i];  
 }  
  
 // Calcular los promedios de cada materia y el promedio total del curso  
 double promedioMatematica = (sumNotasMatematicas/claseMatematicas.length);  
 double promedioHistoria = (sumNotasHistoria/claseHistoria.length);  
 double promedioLenguaje = (sumNotasLenguaje/claseLenguaje.length);  
 System.*out*.println("Promedio clase matemáticas: " + promedioMatematica);  
 System.*out*.println("Promedio clase historia: " + promedioHistoria);  
 System.*out*.println("Promedio clase lenguaje: " + promedioLenguaje);  
 System.*out*.println("Promedio total del curso: "  
 + (promedioMatematica + promedioHistoria + promedioLenguaje)/3);  
  
 // Solicitar al usuario el identificador del alumno y calcular su promedio  
 System.*out*.println("Ingrese el identificador del alumno (de 0 - 6):");  
 int id = s.nextInt();  
 double promedioAlumno = (claseHistoria[id] + claseLenguaje[id] + claseMatematicas[id])/3;  
 System.*out*.println("Promedio alumno Nro " + id + " : " + promedioAlumno);  
 }  
}





# 109. Buscar elementos en un arreglo

import java.util.Scanner;  
  
public class EjemploArreglosBuscarNumero {  
 public static void main(String[] args) {  
  
 int[] a = new int[10]; // Crear el arreglo "a" con capacidad para 10 elementos  
  
 Scanner s = new Scanner(System.*in*);  
  
 for(int i = 0; i < a.length; i++){  
 System.*out*.print("Ingrese un número: ");  
 a[i] = s.nextInt(); // Solicitar al usuario que ingrese los números y almacenarlos en el arreglo "a"  
 }  
  
 System.*out*.println("\r\nIngrese un número a buscar:");  
 int num = s.nextInt(); // Solicitar al usuario el número a buscar  
 int i = 0;  
 // Buscar el número en el arreglo  
 for(; i < a.length && a[i] != num; i++){}  
  
 if(i == a.length){  
 System.*out*.println("Número no encontrado"); // El número no se encontró en el arreglo  
 } else if(a[i] == num){  
 System.*out*.println("Encontrado en la posición: " + i); // El número se encontró en la posición i  
 }  
 }  
}

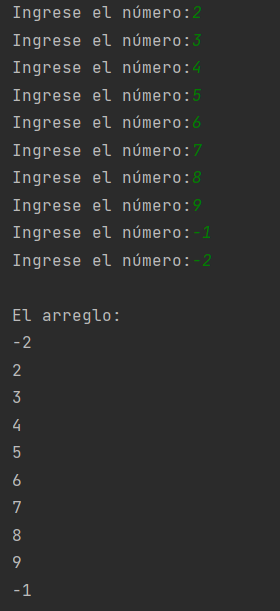


import java.util.Scanner;  
  
public class EjemploArreglosBuscarString {  
 public static void main(String[] args) {  
  
 String[] a = new String[4]; // Crear el arreglo "a" con capacidad para 4 elementos de tipo String  
  
 Scanner s = new Scanner(System.*in*);  
  
 for(int i = 0; i < a.length; i++){  
 System.*out*.print("Ingrese un nombre: ");  
 a[i] = s.next(); // Solicitar al usuario que ingrese los nombres y almacenarlos en el arreglo "a"  
 }  
  
 System.*out*.println("\r\nIngrese un nombre a buscar:");  
 String nombre = s.next(); // Solicitar al usuario el nombre a buscar  
 int i = 0;  
 // Buscar el nombre en el arreglo  
 for(; i < a.length && !a[i].equalsIgnoreCase(nombre); i++){}  
  
 if(i == a.length){  
 System.*out*.println("Nombre no encontrado"); // El nombre no se encontró en el arreglo  
 } else if(a[i].toLowerCase().compareTo(nombre.toLowerCase()) == 0){  
 System.*out*.println("Encontrado en la posición: " + i); // El nombre se encontró en la posición i  
 }  
 }  
}

# 

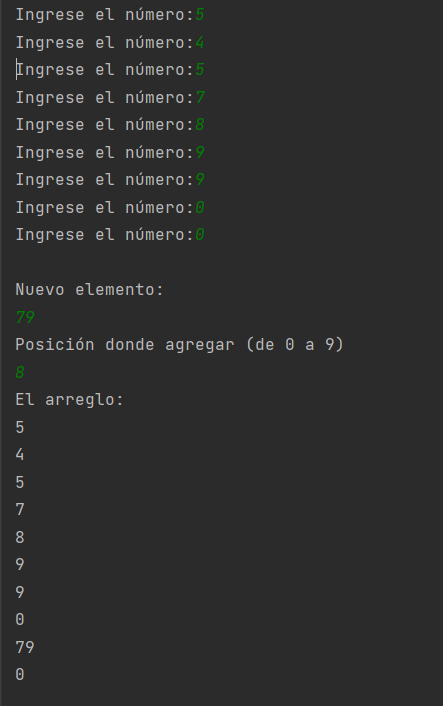
# 110. Mover los elementos en una posición desde el inicio del arreglo hasta el final

import java.util.Scanner;  
  
public class EjemploArreglosDesplazarPosicion {  
 public static void main(String[] args) {  
  
 int[] a = new int[10]; // Crear el arreglo "a" con capacidad para 10 elementos  
 int ultimo;  
 Scanner s = new Scanner(System.*in*);  
  
 for(int i = 0; i < a.length; i++){  
 System.*out*.print("Ingrese el número:");  
 a[i] = s.nextInt(); // Solicitar al usuario que ingrese los números y almacenarlos en el arreglo "a"  
 }  
 System.*out*.println();  
  
 ultimo = a[a.length-1]; // Almacenar el último elemento del arreglo en la variable "ultimo"  
 for(int i = a.length - 2; i >= 0; i--){  
 a[i+1] = a[i]; // Desplazar los elementos hacia la derecha en el arreglo  
 }  
 a[0] = ultimo; // Colocar el último elemento en la posición inicial del arreglo  
  
 System.*out*.println("El arreglo:");  
 for(int i = 0; i < a.length; i++){  
 System.*out*.println(a[i]); // Imprimir los elementos del arreglo  
 }  
  
 }  
}



# 111. Agregar un elemento en una posición indicada y mover los siguientes elementos

//En este código, se solicita al usuario que ingrese 9 números y se almacenan en el arreglo "a". Luego, se solicita al usuario que ingrese un nuevo elemento y la posición donde desea agregarlo en el arreglo.  
//  
// Después de solicitar la entrada del nuevo elemento y la posición, se realiza un desplazamiento de los elementos del arreglo hacia la derecha desde la posición indicada. Luego, se asigna el nuevo elemento en la posición especificada.  
//  
// Finalmente, se imprime el arreglo resultante.  
  
import java.util.Scanner;  
  
public class EjemploArreglosDesplazarPosicion2 {  
 public static void main(String[] args) {  
  
 int[] a = new int[10]; // Crear el arreglo "a" con capacidad para 10 elementos  
 int elemento, posicion;  
 Scanner s = new Scanner(System.*in*);  
  
 for(int i = 0; i < a.length-1; i++){  
 System.*out*.print("Ingrese el número:");  
 a[i] = s.nextInt(); // Solicitar al usuario que ingrese los números y almacenarlos en el arreglo "a"  
 }  
 System.*out*.println();  
  
 System.*out*.println("Nuevo elemento:");  
 elemento = s.nextInt(); // Solicitar al usuario que ingrese el nuevo elemento a agregar  
  
 System.*out*.println("Posición donde agregar (de 0 a 9)");  
 posicion = s.nextInt(); // Solicitar al usuario que ingrese la posición donde agregar el elemento  
  
 for(int i = a.length - 2; i >= posicion; i--){  
 a[i+1] = a[i]; // Desplazar los elementos hacia la derecha desde la posición indicada  
 }  
 a[posicion] = elemento; // Agregar el nuevo elemento en la posición indicada  
  
 System.*out*.println("El arreglo:");  
 for(int i = 0; i < a.length; i++){  
 System.*out*.println(a[i]); // Imprimir los elementos del arreglo  
 }  
  
 }  
}



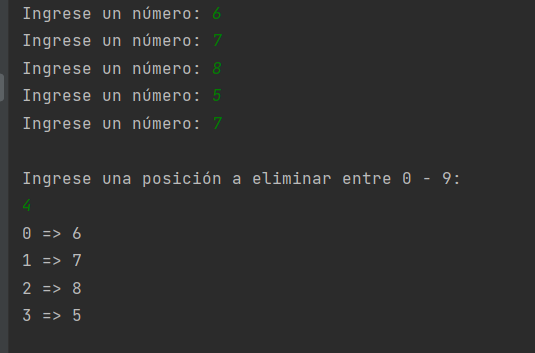
# 112. Agregar un elemento en una posición adecuada de manera que se mantenga… ordenado

/\*  
Enunciado: Agregar un elemento manteniendo el arreglo ordenado  
  
Escriba un programa que solicite al usuario ingresar 6 números enteros y los almacene en un arreglo llamado 'a'.  
Luego, solicite al usuario ingresar un nuevo número a insertar en el arreglo.  
El programa debe insertar el nuevo número en la posición correcta de manera que el arreglo se mantenga  
ordenado de forma ascendente. Finalmente, se debe imprimir el nuevo arreglo ordenado.  
  
\*/  
  
import java.util.Scanner;  
  
public class EjemploArreglosDesplazarPosicion3 {  
 public static void main(String[] args) {  
  
 int[] a = new int[7];  
 int numero, posicion;  
  
 Scanner s = new Scanner(System.*in*);  
  
 for(int i = 0; i < a.length - 1; i++){  
 System.*out*.print("Ingrese un número: ");  
 a[i] = s.nextInt();  
 }  
 System.*out*.println();  
  
 System.*out*.println("Ingrese un número a insertar:");  
 numero = s.nextInt();  
  
 posicion = 0;  
 while(posicion < 6 && numero > a[posicion]){  
 posicion++;  
 }  
  
 for(int i = a.length - 2; i >= posicion; i--){  
 a[i+1] = a[i];  
 }  
  
 a[posicion] = numero;  
 System.*out*.println("El nuevo arreglo ordenado:");  
 for(int i = 0; i < a.length; i++){  
 System.*out*.println(i + " => " + a[i]);  
 }  
  
 }  
}

# 

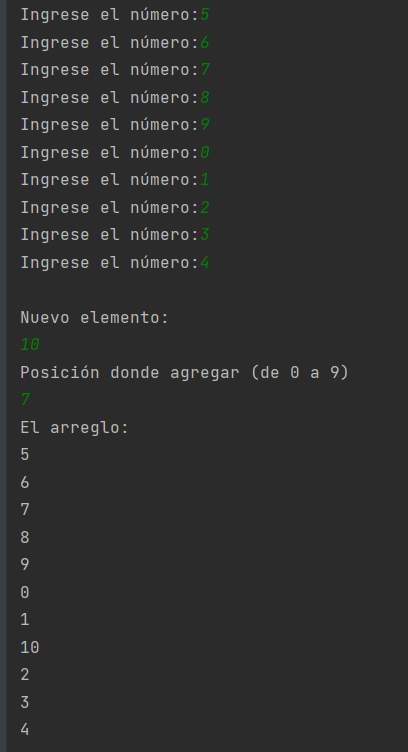
# 113. Eliminar un elemento del arreglo

/\*  
Enunciado: Eliminar un elemento del arreglo  
  
Escriba un programa que solicite al usuario ingresar 10 números enteros y los almacene en un arreglo llamado 'a'.  
Luego, solicite al usuario ingresar una posición (entre 0 y 9) que desea eliminar del arreglo.  
El programa debe eliminar el elemento en la posición indicada y ajustar el tamaño del arreglo.  
Finalmente, se debe imprimir el nuevo arreglo resultante.  
  
\*/  
  
import java.util.Scanner;  
  
public class EjemplosArreglosEliminarElemento {  
 public static void main(String[] args) {  
  
 int[] a = new int[5];  
  
 Scanner s = new Scanner(System.*in*);  
  
 for(int i = 0; i < a.length; i++){  
 System.*out*.print("Ingrese un número: ");  
 a[i] = s.nextInt();  
 }  
 System.*out*.println();  
 System.*out*.println("Ingrese una posición a eliminar entre 0 - 9:");  
 int posicion = s.nextInt();  
  
 for(int i = posicion; i < a.length - 1; i++){  
 a[i] = a[i+1]; // Eliminar el elemento en la posición indicada desplazando los elementos hacia la izquierda  
 }  
  
 int[] b = new int[a.length-1];  
 System.*arraycopy*(a, 0, b, 0, b.length); // Copiar los elementos restantes en un nuevo arreglo de tamaño reducido  
  
 a = b; // Asignar el nuevo arreglo a 'a'  
 for(int i = 0; i<a.length;i++){  
 System.*out*.println(i + " => " + a[i]); // Imprimir el nuevo arreglo resultante  
 }  
 }  
}



# 114. Insertando un nuevo elemento e incrementando el tamaño del arreglo

//En este código, se solicita al usuario que ingrese 10 números y los almacena en el arreglo "a".  
// Luego, se solicita al usuario que ingrese un nuevo elemento y la posición donde desea agregarlo en el arreglo.  
// Después de agregar el elemento en la posición indicada, se crea un nuevo arreglo "b" de tamaño aumentado,  
// se copian los elementos de "a" en "b" y se asigna el nuevo arreglo "b" a "a".  
// Finalmente, se coloca el último elemento en la última posición del arreglo y se imprime el arreglo resultante.  
  
import java.util.Scanner;  
  
public class EjemploArreglosDesplazarPosicion2b {  
 public static void main(String[] args) {  
  
 int[] a = new int[10];  
 int elemento, posicion, ultimo;  
 Scanner s = new Scanner(System.*in*);  
  
 for(int i = 0; i < a.length; i++){  
 System.*out*.print("Ingrese el número:");  
 a[i] = s.nextInt(); // Solicitar al usuario que ingrese los números y almacenarlos en el arreglo "a"  
 }  
 System.*out*.println();  
  
 System.*out*.println("Nuevo elemento:");  
 elemento = s.nextInt(); // Solicitar al usuario que ingrese el nuevo elemento a agregar  
  
 System.*out*.println("Posición donde agregar (de 0 a 9)");  
 posicion = s.nextInt(); // Solicitar al usuario que ingrese la posición donde agregar el elemento  
  
 ultimo = a[a.length -1]; // Almacenar el último elemento del arreglo en la variable "ultimo"  
 for(int i = a.length - 2; i >= posicion; i--){  
 a[i+1] = a[i]; // Desplazar los elementos hacia la derecha desde la posición indicada  
 }  
 int[] b = new int[a.length+1];  
 System.*arraycopy*(a, 0, b, 0, a.length); // Crear un nuevo arreglo "b" de tamaño aumentado  
 a = b; // Asignar el nuevo arreglo "b" a "a"  
 a[posicion] = elemento; // Agregar el nuevo elemento en la posición indicada  
  
 a[a.length -1] = ultimo; // Colocar el último elemento en la última posición del arreglo  
  
 System.*out*.println("El arreglo:");  
 for(int i = 0; i < a.length; i++){  
 System.*out*.println(a[i]); // Imprimir los elementos del arreglo  
 }  
  
 }  
}



# 115. Descargar Código Fuente

# Tarea 13: Ordenar el arreglo de la forma último primero, etc

# Tarea 14: Imprima el numero mas alto del arreglo

# Tarea 15: Sistema estadístico de un arreglo

# Tarea 16: Mayor ocurrencia en el arreglo

# Tarea 17: Imprimir Histograma