

Práctica 5 - Arreglos

1. Cargar un arreglo de tamaño 15, pidiendo el ingreso por teclado de valores entre 1 y 12. Luego mostrar cómo quedó cargado.
2. Hacer un programa que dado un arreglo ya cargado con 10 enteros, calcule el promedio y lo muestre por la consola.
3. Con el mismo arreglo del ejercicio anterior informe por pantalla cuantos elementos del mismo están por encima del promedio calculado.
4. Buscar un elemento en un arreglo de caracteres ya cargado de tamaño 10 y mostrar la/s posición/es del elemento, en caso de no estar indicarlo también.
5. Hacer un programa que dado un arreglo de caracteres de tamaño MAX que se encuentra cargado, invierta el orden del contenido.
6. Hacer un programa que dado un arreglo de enteros de tamaño 10 que se encuentra cargado, obtenga la cantidad de números pares que tiene y la imprima.
7. Hacer un programa que dado un arreglo de enteros y un número N, genere un arreglo con las posiciones donde se encuentra dicho número. A continuación, multiplicar por un número M todas las ocurrencias del número N en el arreglo original.
8. Hacer un programa que determine si los valores almacenados en un arreglo de enteros se encuentran en orden ascendente.
9. Dado un arreglo de caracteres, determinar si en el arreglo existe al menos una letra repetida.
10. Hacer un programa que verifique en un arreglo si tiene al menos N números primos. N debe ser ingresado por el usuario. Reutilizar los métodos ya desarrollados.
11. Dado dos arreglos de números enteros A y B determinar si todos los números almacenados en el arreglo A están presentes en el arreglo B.
12. Dado dos arreglos de números enteros: un arreglo A de tamaño MAXA y un arreglo B de tamaño MAXB con números entre 0 y MAXA-1, determinar la suma de los elementos del arreglo A cuyas posiciones son indicadas por el arreglo B. Por ejemplo, dado $A=\{3,6,31,9\}$, $MAXA=3$ y $B=\{0,2\}$ con $MAXB=2$ el sistema deberá informar: 34 (lo cual es el resultado de sumar el 3 de la posición 0 y el 31 de la posición 2 del arreglo A).
13. Dado un arreglo de caracteres de tamaño MAX, hacer un programa que implemente métodos tal que devuelva al main dos arreglos: un arreglo con los caracteres que son letras vocales y otro con los que son consonantes.

14. Dado un arreglo de caracteres de tamaño MAX, hacer un programa que implemente métodos tal que devuelva al main: un arreglo con las posiciones de los caracteres que son letras vocales y otro con las posiciones que son consonantes.

CORRIMIENTOS

15. Implementar un método que realice un corrimiento a derecha en un arreglo ordenado de tamaño MAX=10 a partir de una posición.
16. Implementar un método que realice un corrimiento a izquierda en un arreglo ordenado de tamaño MAX=10 a partir de una posición.
17. Hacer un programa que dado un arreglo de enteros de tamaño 10 que se encuentra cargado, solicite al usuario un número entero y lo agregue al principio del arreglo (posición 0). Para ello tendrá que realizar un corrimiento a derecha (se pierde el último valor del arreglo) y colocar el número en el arreglo en la posición indicada.
18. Hacer un programa que dado un arreglo de enteros de tamaño 10 que se encuentra precargado, solicite al usuario un número entero y elimine la primera ocurrencia del número (un número igual) en el arreglo (si existe). Para ello tendrá que buscar la posición y si está, realizar un corrimiento a izquierda (queda una copia de la última posición del arreglo en la anteúltima posición).
19. Hacer un programa que dado un arreglo de enteros de tamaño 10 que se encuentra precargado, solicite al usuario un número entero y elimine todas las ocurrencias de número en el arreglo. Mientras exista (en cada iteración tiene que buscar la posición dentro del arreglo) tendrá que usar la posición para realizar un corrimiento a izquierda (quedarán tantas copias de la última posición del arreglo como cantidad de ocurrencias del número).
20. Suponer a partir de lo resuelto en el ejercicio anterior (11) que el elemento a eliminar coincide con el último que hay en el arreglo. ¿Qué pasa en este caso? ¿Cómo daría una solución al problema?
21. Hacer un programa que inserte un elemento en un arreglo (ordenado decrecientemente) de tamaño MAX=10.
22. Hacer un programa que elimine los valores pares en un arreglo de tamaño MAX=10.

SECUENCIAS

Dado un arreglo de tamaño de arreglo MAX=20 de secuencias. En el caso de eliminar secuencias se debe hacer corrimiento.

23. Hacer un programa que devuelva la posición de inicio y fin de la

- primera secuencia de números distinta de ceros.
24. Hacer un programa que devuelva la posición de inicio y fin de la secuencia de números distintos de ceros cuya suma del contenido sea mayor.
 25. Hacer un programa que devuelva la posición de inicio y fin de la anteúltima secuencia de números distintos de ceros.
 26. Hacer un programa que devuelva la posición de inicio y de fin de la secuencia de mayor tamaño.
 27. Hacer un programa que dado un número N ingresado por el usuario, elimine las secuencias de tamaño N de números distintos de cero.
 28. Hacer un programa que elimine de un arreglo todas las ocurrencias de una secuencia patrón dada por otro arreglo.
 29. Hacer un programa que elimine de un arreglo todas las secuencias que tienen orden descendente entre sus elementos.
 30. Hacer un programa que reemplace de un arreglo A todas las ocurrencias de una secuencia patrón dada en un arreglo P, por la secuencia contenida en el arreglo R de igual tamaño.
 31. Hacer un programa que invierta el orden de la última secuencia en un arreglo.
 32. Se tiene un arreglo de enteros de tamaño 20 de secuencias de números entre 1 y 9, separadas por 0. El arreglo está precargado, y además empieza y termina con uno o más separadores 0. Hacer un programa que permita obtener a través de métodos la posición de inicio y la posición de fin de la secuencia ubicada a partir de una posición entera ingresada por el usuario. Finalmente, si existen imprima por pantalla ambas posiciones obtenidas.
 33. Hacer un programa que dado el arreglo definido y precargado, y un número entero ingresado por el usuario, copie de forma continua las secuencias de tamaño igual al número ingresado en otro arreglo de iguales características e inicializado con 0. La copia en este último arreglo deben comenzar desde el principio del mismo.
 34. Suponer que se tienen dos arreglos A y B de secuencias (de caracteres separados por uno o más espacio) de tamaño MAX (cada arreglo empieza y termina con uno o más caracteres espacio). A y B están precargados y además se tienen los siguientes métodos (existen y no se tienen que implementar):
 - a. un método que permite obtener el índice inicial de la secuencia que más se repite de un arreglo de secuencias de tamaño MAX (que empieza y termina con o más caracteres espacios).
 - b. un método que retorna el índice inicial de la secuencia que

tiene más caracteres repetidos de un arreglo de secuencias (de caracteres separados por uno o más caracteres espacio) de tamaño MAX (que empieza y termina con espacio).

Se pide realizar un programa completo que (sin utilizar arreglos auxiliares/extras a los mencionados):

_reemplace en A la secuencia que más se repite por la secuencia de B con más caracteres repetidos (la separación previa entre las secuencias de A debe mantenerse). Considerar poner un separador al final del arreglo si se desplaza a derecha para mantener consistencia en la terminación de la secuencia.

En caso de utilizar uno o más de los métodos preexistentes mencionados en el enunciado definir el o los encabezados de los mismos. No se requiere hacer métodos de carga e impresión de arreglos.

ORDENAMIENTO

No hay que aplicar un método de ordenamiento cuando indica que el arreglo se encuentra ordenado.

35. Hacer un programa que dado un arreglo ordenado creciente de enteros de tamaño 10 que se encuentra precargado, solicite al usuario un número entero y lo inserte en el arreglo manteniendo su orden. Para ello tendrá que realizar un corrimiento a derecha (se pierde el último valor del arreglo) y colocar el número en el arreglo en la posición indicada.
36. Hacer un programa que dado un arreglo ordenado creciente de enteros de tamaño 10 que se encuentra precargado, solicite al usuario un número entero y elimine la primera ocurrencia de número (un número igual) en el arreglo (si existe). Informar si no existe en el arreglo.
37. Hacer un programa que dado un arreglo de enteros de tamaño 10 que se encuentra precargado, solicite al usuario el ingreso de dos números enteros (posiciones del arreglo) y ordene de forma creciente el arreglo entre dos posiciones correspondientes a los números ingresados.
38. Hacer un programa que cargue un arreglo (sin orden) de tamaño MAX=10 con números enteros aleatorios entre 0 y 100. Finalmente imprima por pantalla el arreglo ordenado. Hacerlo para orden creciente y decreciente

Bonus Track 1 - Ejercicio dominio real

1. Un sismógrafo ubicado en la ciudad de Mendoza, Argentina, registró una fuerte actividad sísmica en los últimos días. Los datos están en escala de Richter, y están representados en el siguiente arreglo:

0	2.1	1.5	4.1	0	3.4	0	2.5	3.0	0	3.8	0	2.7	2.7	3.9	2.5	0	4.2	2.0	0
---	-----	-----	-----	---	-----	---	-----	-----	---	-----	---	-----	-----	-----	-----	---	-----	-----	---

Cada día de actividad sísmica está separada por un cero. Cada secuencia representa qué sismos hubo en esa jornada. Entonces, con la información provista, se pide calcular el promedio de actividad sísmica para aquellos días donde hubo un número mayor a N de temblores y que todas las magnitudes de dicho día sean de un mínimo de M en la escala de Richter

Por ejemplo, si $N=1$ y $M=2.0$, las secuencias que cumplen son:

0	2.1	1.5	4.1	0	3.4	0	2.5	3.0	0	1.8	0	2.7	2.7	3.9	2.5	0	4.2	2.0	0
---	-----	-----	-----	---	-----	---	-----	-----	---	-----	---	-----	-----	-----	-----	---	-----	-----	---

Por ende, el promedio de sismos para los días que cumplen la condición es $\Rightarrow 2.9375$

2. El departamento de ventas de la cadena de supermercados carrefive desea hacer un análisis sobre lo comercializado en el último fin de semana largo. Para ello solicita al grupo de desarrollo que le prepare algunas rutinas para conocer:
- cuántos tickets se emitieron ese fin de semana
 - cuánto fue el importe total de ventas
 - cuál fue el ticket de mayor importe, cuánto fue su valor
 - cuál fue la mayor cantidad de artículos vendidos en un ticket
 - del ticket con mayor cantidad de artículos vendidos, cuál fue el de mayor importe, el de menor y el promedio
 - del ticket de mayor importe, cuántos artículos hay involucrados y cuál es el promedio de valores
 - cuanto es el valor promedio de venta por ticket en todo el fin de semana
 - cuál fue el ticket con el menor importe, cuánto fue su valor
 - cuál fue el ticket con la menor cantidad de artículos vendidos
- Hacer un programa completo con un menú de opciones (con la opción '0' sale) para que informe los diferentes valores al

departamento de ventas. Considerar constantes y variables acordes al problema. Establezca cómo será la estructura que usará y defínala en el programa teniendo en cuenta los conceptos de secuencias vistos en la teoría.

3. Dado que el equipo de desarrollo es muy eficiente, el departamento de ventas le ha generado un nuevo requerimiento para que el sistema pueda emitir algunos informes extras:
 - a. dado un importe, el sistema le debe informar cuantos tickets superan dicho importe
 - b. dada una cantidad de artículos, el sistema le debe informar cuántos tickets están por encima de esa cantidad.
 - c. dado un importe, el sistema debe informar cuantos artículos vendidos están por encima del mismo.
 - d. dado un importe promedio, cuántos tickets superan en promedio a dicho importe

Hacer un programa completo con un menú de opciones (con la opción '0' sale) para que informe los diferentes valores al departamento de ventas.

4. Una refactorización del sistema se ha aplicado sobre la estructura que permite almacenar los tickets para poder soportar una nueva funcionalidad. En el nuevo esquema cada secuencia tiene en primer lugar el id del producto, al lado la cantidad vendida de ese producto para ese ticket. Ese se repite hasta que no hay más productos y el último lugar lo ocupa el importe total. Ejemplo:

0	3	24	7	1	203.56	0	99	2	87.5	0	12	8	3	5	1233.4	0
---	---	----	---	---	--------	---	----	---	------	---	----	---	---	---	--------	---

Para este ejemplo tenemos 3 tickets:

- i. el primero para el producto de id 3 se vendieron 24, para el id 7, 1 solo y el importe de la venta fue de \$203.56
- ii. para el segundo solo se vendieron 2 productos del id 99 por un importe de \$87.5
- iii. para el tercero se vendieron 8 del id 12 y 5 del id 3 todo por un importe de \$1233.4

Con este esquema el departamento de ventas pretende conocer cuantos descuentos se aplicaron sabiendo que si la cantidad de productos iguales es mayor o igual a 2 y es una cantidad par se aplicó un descuento del 10% en el valor total del ticket. Tomando en cuenta estos descuentos el departamento de ventas quiere saber

- a. cuánto fue el total de descuentos aplicados en todo el fin de

semana para lo cual necesita saber cuánto sumó en pesos todos los descuentos aplicados. Para el ejemplo suma: \$152.45 (redondeado)

- b. cuantos productos tuvieron descuento (la cantidad de productos diferentes no la cantidad de unidades). Para el ejemplo fueron 3 artículos: el de id 3, id 99 e id 12.

5. El algoritmo de Luhn o fórmula de Luhn, es una fórmula de suma de verificación, utilizada para validar una diversidad de números de identificación; como números de tarjetas de crédito, números IMEI, etc. Implementar un programa que determine si un número almacenado digito a digito (de 0 a 9) en un arreglo de tamaño 16 corresponde a un número correcto según el algoritmo de Luhn. Dicho algoritmo consiste en duplicar cada dígito en posición par comenzando de derecha a izquierda. Por ejemplo, dado el siguiente arreglo de dígitos:

7	9	9	2	7	3	9	8	7	1	3
---	----------	---	----------	---	----------	---	----------	---	----------	---

Se debe comenzar de derecha a izquierda con el 1 y duplicarlo, luego con el 8, 3, 2 y 9 (en negrita). Luego de este primer paso el arreglo resultante será:

7	18	9	4	7	6	9	16	7	2	3
---	-----------	---	----------	---	----------	---	-----------	---	----------	---

Como segundo paso, si alguno de los números duplicados tuviese dos dígitos se los debe sumar y obtener así un solo dígito:

7	9	9	4	7	6	9	7	7	2	3
---	----------	---	----------	---	----------	---	----------	---	----------	---

Por último, se debe sumar todos los valores del arreglo ($7+9+9+4+7+6+9+7+7+2+3=70$). Si dicha suma es múltiplo de 10 el número es correcto, caso contrario es incorrecto.

Bonus Track 2 - Ejercicio análisis de código

1. Dados los siguientes códigos, analizar y detectar los errores. Justificar en cada caso.

a.

```
public class Practica_5_Bonus_1 {
    public static final int MAX=10;
    public static void main (String [] args){
        int[] arreglo= new int[MAX];
        int pos=6;
```

```

        correrDerecha(arreglo, pos);
        mostrarArreglo(arreglo); //ya implementado
    }
    public static void correrDerecha(int[] arr, int pos){
        for (int i = 0 ; i <= MAX; i++) {
            arr[i]=arr[i-1];
        }
    }
}

```

b.

```

public class Practica_5_Bonus_2 {
    public static final int MAX=10;
    public static void main (String [] args){
        int[] arreglo= new int[MAX];
        int pos=6;
        correrIzquierda(arreglo, pos);
        mostrarArreglo(arreglo); //ya implementado
    }
    public static void correrIzquierda(int[] arr, int pos){
        for (int i = 0 ; i <= MAX; i++) {
            arr[i]=arr[i+1];
        }
    }
}

```

C.

```

public class Practica_5_Bonus_3 {
    public static final int MAX=10;
    public static void main (String [] args){
        int[] arreglo= new int[MAX];
        int numero=6;
        int pos=buscarPosNumero(arreglo, numero);
        System.out.println("La posicion del numero: " +
        numero + " es:" + pos);
    }
    public static void buscarPosNumero(int[] arr, int
    numero){
        for (int i = 0 ; i < MAX; i++){
            if (arr[i]== numero)
                return i;
        }
        return -1;
    }
}

```

2. Dados los siguientes enunciados y códigos, analizar y detectar los errores. Justificar en cada caso.

a. Hacer un programa que dado un arreglo de enteros de tamaño 10 que se encuentra precargado, imprima por pantalla el promedio de la suma de sus valores.

```

public class Practica_5_Bonus_4 {
    public static void main (String [] args) {

```



```

        public static int MAX = 10;
        char [] arrenteros = new int [MAX];
        int promedio;
        cargar_arreglo_int(arrenteros); //ya implementado
        promedio = promedio_arreglo(arrenteros);
        System.out.println("El promedio del arreglo es: " +
            promedio);
    }
    public static int promedio_arreglo(int [] arr){
        int suma = 0;
        for (int pos = 0; pos <= MAX; pos++){
            suma+=arr[pos];
        }
        return (suma/MAX);
    }
}

```

- b. Escribir un programa que elimine todas las secuencias de tamaño par.

```

public class Practica_5_Bonus_5 {
    final static int MAX = 20;
    public static void main (String[] args) {
        int[] A=new int[MAX], B=new int[MAX];
        B=cargar_arreglo(A);
        eliminarSecuencias(B);
    }
    public static int[] cargar_arreglo(int[] arr) {
        for (int pos = 0 ; pos < MAX; pos++){
            arr[pos]=pos*2;
        }
        return arr;
    }
    public static void eliminarSecuencias(int[] arr){
        int ini=0,fin=-1;
        while (fin<MAX){
            ini=buscarIni(arr, fin+1); // ya implementado
            if (ini<MAX){
                fin=buscarFin(arr,ini); // ya
implementado
                int tam=fin-ini+1;
                if (tam%2==0){
                    for (; tam>0;tam-){
                        correrIzq(arr,ini); // ya
implementado
                    }
                }
                fin++;
            }
        }
    }
    public static int obtenerIni(int[] arr, int ini) {
        while (arr[ini] == 0 && ini<MAX)
            ini++;
        return ini;
    }
    public static int obtenerFin(int[] arr, int ini) {
        while (arr[ini] !=' ' && ini<MAX)
            ini++;
    }
}

```

```
        return ini-1;
    }
}
```