

Programación en Java

Ejercicios de estructuras de control y arrays

Ejercicios:

1. Escribir un bucle equivalente al del siguiente programa sin utilizar el operador &&.

```
int lim = 10;
int c = 1;
Scanner sc = new Scanner(System.in);
for (int i = 0; i < lim && c != 0; i++, c = sc.nextInt()) {}
```

2. Escriba un programa que lea de la entrada un número y determine si es un número primo.
3. Escriba un programa que sume todos los elementos de un array de enteros. Utilícelo para sumar el vector:
`int[] a = {1, 2, 3, 4, 5, 6};`
4. Hacer un programa que dados dos arrays determine si son iguales. Probarlo con los arrays [1,2,3] y [1,2,3], [1,2,3,4] y [1,2,3] y [1,2,3] y [1,2,2]
5. Escribir un programa que recoja de la entrada estándar dos arrays de enteros de tamaño especificado por el usuario y los multiplique dos a dos, guardando el resultado en otro array e imprimiendo los valores de este último array.
6. Escriba un programa que lea de la entrada un número y utilice ese número para crear un array de enteros de ese tamaño. A continuación, el programa inicializará el array con los valores que el usuario introduzca por la entrada. Finalmente, se deberá recorrer el array e imprimir tantos caracteres '*' como indique el valor de cada elemento del array. Ejemplo: los valores 5,4,3,2 producirán la siguiente salida:

```
*****
****
***
**
```

7. Declarar e inicializar una matriz de enteros de 2x3 e imprimirla.
8. Escribir un programa que calcule el determinante de una matriz de 3x3. Pruébalo con la matriz [[1,2,3], [4,5,6], [7,8,8]] cuyo determinante es 3

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$\det A = aei + bfg + cdh - afh - bdi - ceg.$$

Ejercicios Extra

9. Escribir un programa que lea un número de enteros entre 1 y 100 de la entrada estándar y genere un histograma con las frecuencias de cada entero en la secuencia. Para representar las barras del histograma utilizar secuencias del carácter '*'. Por ejemplo, la secuencia: 1, 1, 20, 1, 2, 20, 3, 3, 3,4,4,4,33,3 generaría la siguiente salida:

```
1: ***
2: *
3: ****
4: ***
20:**
33:*
```

10. Escribe un programa que dada una matriz de 3x3 realice las siguientes funciones:

- Calcule la media de todos los valores
- Calcule el valor mínimo y el máximo de todos los valores
- Sumar los valores en la diagonal descendente.
- Sumar los valores en la diagonal ascendente.
- Sumar los valores que están por encima de la diagonal descendente.
- Sumar los valores que están por debajo de la diagonal descendente.
- Probarlo con la matriz [[1,2,3], [4,5,6], [7,8,9]]

11. Escriba un programa simule varios lanzamientos de un dado y muestre la frecuencia de cada una de las caras así como la proporción del número de veces que haya salido cada cara con respecto al total. Para ello simule 6000 lanzamientos de un dado utilizando la función Math.random()*6 para simular cada lanzamiento.

12. Escribe un programa que ordene los números de un vector. Utilizar el algoritmo de la burbuja y probarlo sobre los vectores [4,3,2,1] y [3,2,1,4]. Algoritmo de la burbuja (bubblesort) en pseudocódigo

```
for i from 1 to n
  for j from i to 2
    if (A[j] < A[j - 1])
      intercambiar los valores entre A[j] y A[j - 1]
```

13. Escriba un programa que, dado un array de enteros, busque si un número obtenido por la entrada estándar se encuentra en el array e imprima la primera posición en la que se encuentra el número.

14. Implementar el algoritmo de búsqueda binaria sobre un array/vector de enteros **ordenado**. El programa debe devolver si ha encontrado o no el número y el número de comparaciones con elementos del vector que ha requerido para encontrarlo. Probarlo con el vector

```
int v[] = {2,3,4,5,6,7,8,9,10,11,12} y los números 11 y 13
```

```

Algoritmo binario en pseudocódigo
menor_pos = 0
mayor_pos = N
medio_pos = (menor_pos + mayor_pos) / 2
while (menor_pos <= mayor_pos AND vector[medio_pos] != valor) {
    if (valor < vector[medio_pos])
        mayor_pos = medio_pos - 1;
    else
        menor_pos = medio_pos + 1;
    medio_pos = (menor_pos + mayor_pos) / 2
}

```

15. Escriba un programa que pida un número al usuario y muestre por pantalla el resultado de su raíz cuadrada. Para realizar este ejercicio se puede utilizar un método aproximado que permita llegar a la solución con un error menor de 0,0001. Lógicamente para su resolución no se puede utilizar la función de biblioteca Math.sqrt() pero sí se puede utilizar la que devuelve el valor absoluto de un real (Math.abs(numero)).