

A partir de ahora comenzaremos a aprender cómo los mismos algoritmos que diseñamos en PSeInt podemos escribirlos también en Java, simplemente haciendo una traducción de cada una de las estructuras de control vistas en PSeInt a Java.

Si bien en esta guía se proponen nuevos problemas, se sugiere que los mismos ejercicios ya implementados en PSeInt sean traducidos al lenguaje de programación Java.

También recordamos que las resoluciones de nuestros ejercicios debemos subirlas a un repositorio propio de GitHub para seguir practicando el uso de Git y GitHub



**VIDEOS:** Te sugerimos ver los videos relacionados con este tema, antes de empezar los ejercicios, los podrás encontrar en tu aula virtual o en nuestro canal de YouTube.

Los ejercicios van a seguir con el siguiente filtro de dificultad:

Dificultad Baja

Dificultad Media

Dificultad Alta

1. Escribir un programa que pida dos números enteros por teclado y calcule la suma de los dos. El programa deberá después mostrar el resultado de la suma
2. Escribir un programa que pida tu nombre, lo guarde en una variable y lo muestre por pantalla.
3. Escribir un programa que pida una frase y la muestre toda en mayúsculas y después toda en minúsculas. **Nota: investigar la función `toUpperCase()` y `toLowerCase()` en Java.**
4. Dada una cantidad de grados centígrados se debe mostrar su equivalente en grados Fahrenheit. La fórmula correspondiente es:  $F = 32 + (9 * C / 5)$ .
5. Escribir un programa que lea un número entero por teclado y muestre por pantalla el doble, el triple y la raíz cuadrada de ese número. **Nota: investigar la función `Math.sqrt()`.**

### Condicionales en Java

6. Crear un programa que dado un numero determine si es par o impar.
7. Crear un programa que pida una frase y si esa frase es igual a "eureka" el programa pondrá un mensaje de Correcto, sino mostrará un mensaje de Incorrecto. **Nota: investigar la función `equals()` en Java.**
8. Realizar un programa que solo permita introducir solo frases o palabras de 8 de largo. Si el usuario ingresa una frase o palabra de 8 de largo se deberá de imprimir un mensaje por pantalla que diga "CORRECTO", en caso contrario, se deberá imprimir "INCORRECTO". **Nota: investigar la función `Length()` en Java.**

9. Escriba un programa que pida una frase o palabra y valide si la primera letra de esa frase es una 'A'. Si la primera letra es una 'A', se deberá de imprimir un mensaje por pantalla que diga "CORRECTO", en caso contrario, se deberá imprimir "INCORRECTO". Nota: investigar la función Substring y equals() de Java.

### Bucles y sentencias de salto break y continue

10. Escriba un programa en el cual se ingrese un valor límite positivo, y a continuación solicite números al usuario hasta que la suma de los números introducidos supere el límite inicial.
11. Realizar un programa que pida dos números enteros positivos por teclado y muestre por pantalla el siguiente menú:

```
MENU
1. Sumar
2. Restar
3. Multiplicar
4. Dividir
5. Salir
Elija opción:
```

El usuario deberá elegir una opción y el programa deberá mostrar el resultado por pantalla y luego volver al menú. El programa deberá ejecutarse hasta que se elija la opción 5. Tener en cuenta que, si el usuario selecciona la opción 5, en vez de salir del programa directamente, se debe mostrar el siguiente mensaje de confirmación: ¿Está seguro que desea salir del programa (S/N)? Si el usuario selecciona el carácter 'S' se sale del programa, caso contrario se vuelve a mostrar el menú.

12. Realizar un programa que simule el funcionamiento de un dispositivo RS232, este tipo de dispositivo lee cadenas enviadas por el usuario. Las cadenas deben llegar con un formato fijo: tienen que ser de un máximo de 5 caracteres de largo, el primer carácter tiene que ser X y el último tiene que ser una O.

Las secuencias leídas que respeten el formato se consideran correctas, la secuencia especial "&&&&&" marca el final de los envíos (llamémosla FDE), y toda secuencia distinta de FDE, que no respete el formato se considera incorrecta.

Al finalizar el proceso, se imprime un informe indicando la cantidad de lecturas correctas e incorrectas recibidas. Para resolver el ejercicio deberá investigar cómo se utilizan las siguientes funciones de Java **Substring()**, **Length()**, **equals()**.

13. Dibujar un cuadrado de N elementos por lado utilizando el carácter "\*". Por ejemplo, si el cuadrado tiene 4 elementos por lado se deberá dibujar lo siguiente:

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

\*   \*  
\* \* \* \*

## Vectores y Matrices en Java

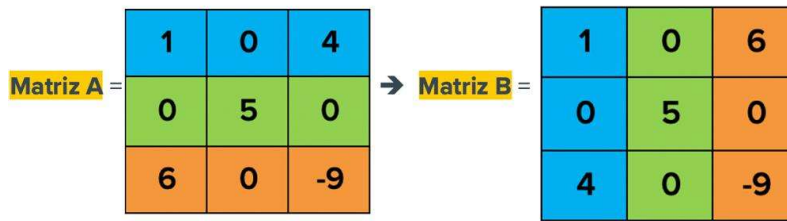
14. Crea una aplicación que a través de una función nos convierta una cantidad de euros introducida por teclado a otra moneda, estas pueden ser a dólares, yenes o libras. La función tendrá como parámetros, la cantidad de euros y la moneda a converir que será una cadena, este no devolverá ningún valor y mostrará un mensaje indicando el cambio (void).

El cambio de divisas es:

- \* 0.86 libras es un 1 €
- \* 1.28611 \$ es un 1 €
- \* 129.852 yenes es un 1 €

## Funciones en Java

15. Realizar un algoritmo que rellene un vector con los 100 primeros números enteros y los muestre por pantalla en orden descendente.
16. Realizar un algoritmo que rellene un vector de tamaño N con valores aleatorios y le pida al usuario un numero a buscar en el vector. El programa mostrará donde se encuentra el numero y si se encuentra repetido
17. Recorrer un vector de N enteros contabilizando cuántos números son de 1 dígito, cuántos de 2 dígitos, etcétera (hasta 5 dígitos).
18. Realizar un programa que rellene un matriz de 4 x 4 de valores aleatorios y muestre la traspuesta de la matriz. La matriz traspuesta de una matriz A se denota por B y se obtiene cambiando sus filas por columnas (o viceversa).



19. Realice un programa que compruebe si una matriz dada es anti simétrica. Se dice que una matriz A es anti simétrica cuando ésta es igual a su propia traspuesta, pero cambiada de signo. Es decir, A es anti simétrica si  $A = -A^T$ . La matriz traspuesta de una matriz A se denota por  $A^T$  y se obtiene cambiando sus filas por columnas (o viceversa).

Matriz

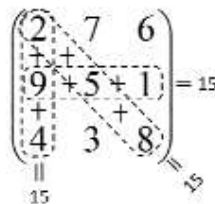
0	-2	4
2	0	2
-4	-2	0

Matriz Transpuesta

0	2	-4
-2	0	-2
4	2	0

**En este caso la matriz es anti simétrica.**

20. Un cuadrado mágico 3 x 3 es una matriz 3 x 3 formada por números del 1 al 9 donde la suma de sus filas, sus columnas y sus diagonales son idénticas. Crear un programa que permita introducir un cuadrado por teclado y determine si este cuadrado es mágico o no. El programa deberá comprobar que los números introducidos son correctos, es decir, están entre el 1 y el 9.



21. Dadas dos matrices cuadradas de números enteros, la matriz M de 10x10 y la matriz P de 3x3, se solicita escribir un programa en el cual se compruebe si la matriz P está contenida dentro de la matriz M. Para ello se debe verificar si entre todas las submatrices de 3x3 que se pueden formar en la matriz M, desplazándose por filas o columnas, existe al menos una que coincida con la matriz P. En ese caso, el programa debe indicar la fila y la columna de la matriz M en la cual empieza el primer elemento de la submatriz P.

Ejemplo:

Matriz de 10 x 10

1	26	36	47	5	6	72	81	95	10
11	12	13	21	41	22	67	20	10	61
56	78	87	90	09	90	17	12	87	67
41	87	24	56	97	74	87	42	64	35
32	76	79	1	36	5	67	96	12	11
99	13	54	88	89	90	75	12	41	76
67	78	87	45	14	22	26	42	56	78
98	45	34	23	32	56	74	16	19	18
24	67	97	46	87	13	67	89	93	24
21	68	78	98	90	67	12	41	65	12

Matriz de 3 x 3

36	5	67
89	90	75
14	22	26

Como podemos observar nuestra submatriz P se encuentra en la matriz M en los índices: 4,4 - 4,5 - 4,6 - 5,4 - 5,5 - 5,6 - 6,4 - 6,5 - 6,6.

## EJERCICIOS DE APRENDIZAJE EXTRA

Estos van a ser ejercicios para reforzar los conocimientos previamente vistos. Estos pueden realizarse cuando hayas terminado la guía y tengas una buena base sobre lo que venimos trabajando. Además, si ya terminaste la guía y te queda tiempo libre en las mesas, puedes continuar con estos ejercicios extra, recordando siempre que no es necesario que los termines para continuar con el tema siguiente. Por último, recordá que la prioridad es ayudar a los compañeros de la mesa y que cuando tengas que ayudar, lo más valioso es que puedas explicar el ejercicio con la intención de que tu compañero lo comprenda, y no sólo mostrarlo. ¡Muchas gracias!

1. Dado un tiempo en minutos, calcular su equivalente en días y horas. Por ejemplo, si el usuario ingresa 1600 minutos, el sistema debe calcular su equivalente: 1 día, 2 horas.
2. Declarar cuatro variables de tipo entero **A**, **B**, **C** y **D** y asignarle un valor diferente a cada una. A continuación, realizar las instrucciones necesarias para que: **B** tome el valor de **C**,

**C** tome el valor de **A**, **A** tome el valor de **D** y **D** tome el valor de **B**. Mostrar los valores iniciales y los valores finales de cada variable. Utilizar sólo una variable auxiliar.

3. Elaborar un algoritmo en el cuál se ingrese una letra y se detecte si se trata de una vocal. Caso contrario mostrar un mensaje. **Nota: investigar la función equals() de la clase String.**

4. Elaborar un algoritmo en el cuál se ingrese un número entre 1 y 10 y se muestre su equivalente en romano.

5. Una obra social tiene tres clases de socios:

- Los socios tipo 'A' abonan una cuota mayor, pero tienen un 50% de descuento en todos los tipos de tratamientos.
- Los socios tipo 'B' abonan una cuota moderada y tienen un 35% de descuento para los mismos tratamientos que los socios del tipo A.
- Los socios que menos aportan, los de tipo 'C', no reciben descuentos sobre dichos tratamientos.
- Solicite una letra (carácter) que representa la clase de un socio, y luego un valor real que represente el costo del tratamiento (previo al descuento) y determine el importe en efectivo a pagar por dicho socio.

6. Leer la altura de N personas y determinar el promedio de estaturas que se encuentran por debajo de 1.60 mts. y el promedio de estaturas en general.

7. Realice un programa que calcule y visualice el valor máximo, el valor mínimo y el promedio de n números ( $n > 0$ ). El valor de n se solicitará al principio del programa y los números serán introducidos por el usuario. Realice dos versiones del programa, una usando el bucle "while" y otra con el bucle "do - while".

8. Escriba un programa que lea números enteros. Si el número es múltiplo de cinco debe detener la lectura y mostrar la cantidad de números leídos, la cantidad de números pares y la cantidad de números impares. Al igual que en el ejercicio anterior los números negativos no deben sumarse. **Nota: recordar el uso de la sentencia break.**

9. Simular la división usando solamente restas. Dados dos números enteros mayores que uno, realizar un algoritmo que calcule el cociente y el residuo usando sólo restas. Método: Restar el dividendo del divisor hasta obtener un resultado menor que el divisor, este resultado es el residuo, y el número de restas realizadas es el cociente.

Por ejemplo:  $50 / 13$ :

$50 - 13 = 37$  una resta realizada

$37 - 13 = 24$  dos restas realizadas

$24 - 13 = 11$    tres restas realizadas

dado que 11 es menor que 13, entonces: el residuo es 11 y el cociente es 3.

¿Aún no lo entendiste? **Recomendamos googlear división con restas sucesivas.**

10. Realice un programa para que el usuario adivine el resultado de una multiplicación entre dos números generados aleatoriamente entre 0 y 10. El programa debe indicar al usuario si su respuesta es o no correcta. En caso que la respuesta sea incorrecta se debe permitir al usuario ingresar su respuesta nuevamente. Para realizar este ejercicio investigue como utilizar la función **Math.random()** de Java.

11. Escribir un programa que lea un número entero y devuelva el número de dígitos que componen ese número. Por ejemplo, si introducimos el número 12345, el programa deberá devolver 5. Calcular la cantidad de dígitos matemáticamente utilizando el operador de división. **Nota: recordar que las variables de tipo entero truncan los números o resultados.**

12. Necesitamos mostrar un contador con 3 dígitos (X-X-X), que muestre los números del 0-0-0 al 9-9-9, con la particularidad que cada vez que aparezca un 3 lo sustituya por una E. Ejemplo:

0-0-0

0-0-1

0-0-2

0-0-E

0-0-4

0-1-2

0-1-E

**Nota: investigar función equals() y como convertir números a String.**

13. Crear un programa que dibuje una escalera de números, donde cada línea de números comience en uno y termine en el número de la línea. Solicitar la altura de la escalera al usuario al comenzar. Ejemplo: si se ingresa el número 3:

1

12

123

14. Se dispone de un conjunto de N familias, cada una de las cuales tiene una M cantidad de hijos. Escriba un programa que pida la cantidad de familias y para cada familia la cantidad de hijos para averiguar la media de edad de los hijos de todas las familias.

15. Crea una aplicación que le pida dos números al usuario y este pueda elegir entre sumar, restar, multiplicar y dividir. La aplicación debe tener una función para cada operación matemática y deben devolver sus resultados para imprimirlos en el main.

16. Diseñe una función que pida el nombre y la edad de N personas e imprima los datos de las personas ingresadas por teclado e indique si son mayores o menores de edad. Después de cada persona, el programa debe preguntarle al usuario si quiere seguir mostrando personas y frenar cuando el usuario ingrese la palabra "No".

17. Crea una aplicación que nos pida un número por teclado y con una función se lo pasamos por parámetro para que nos indique si es o no un número primo, debe devolver true si es primo, sino false.

Un número primo es aquel solo puede dividirse entre 1 y si mismo. Por ejemplo: 25 no es primo, ya que 25 es divisible entre 5, sin embargo, 17 si es primo.

### ¿Qué son los números primos?

Básicamente, un número primo es un **número natural que tiene solo dos divisores o factores**: 1 y el mismo número. Es decir, es primo aquel número que se puede dividir por uno y por el mismo número.

El primer número primo es 2, y hay 25 números primos entre 1 y 100, ellos son: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 y 97.

18. Realizar un algoritmo que calcule la suma de todos los elementos de un vector de tamaño N, con los valores ingresados por el usuario.

19. Escriba un programa que averigüe si dos vectores de N enteros son iguales (la comparación deberá detenerse en cuanto se detecte alguna diferencia entre los elementos).

20. Crear una función rellene un vector con números aleatorios, pasándole un arreglo por parámetro. Después haremos otra función o procedimiento que imprima el vector.

21. Los profesores del curso de programación de Egg necesitan llevar un registro de las notas adquiridas por sus 10 alumnos para luego obtener una cantidad de aprobados y desaprobados. Durante el periodo de cursado cada alumno obtiene 4 notas, 2 por trabajos prácticos evaluativos y 2 por parciales. Las ponderaciones de cada nota son:

**Primer trabajo práctico evaluativo 10%**

**Segundo trabajo práctico evaluativo 15%**

**Primer Integrador 25%**

**Segundo integrador 50%**

Una vez cargadas las notas, se calcula el promedio y se guarda en el arreglo. Al final del programa los profesores necesitan obtener por pantalla la cantidad de aprobados y



desaprobados, teniendo en cuenta que solo aprueban los alumnos con promedio mayor o igual al 7 de sus notas del curso.

22. Realizar un programa que rellene una matriz de tamaño NxM con valores aleatorios y muestre la suma de sus elementos.

23. Construya un programa que lea 5 palabras de mínimo 3 y hasta 5 caracteres y, a medida que el usuario las va ingresando, construya una "sopa de letras para niños" de tamaño de 20 x 20 caracteres. Las palabras se ubicarán todas en orden horizontal en una fila que será seleccionada de manera aleatoria. Una vez concluida la ubicación de las palabras, rellene los espacios no utilizados con un número aleatorio del 0 al 9. Finalmente imprima por pantalla la sopa de letras creada.

Nota: Para resolver el ejercicio deberá investigar cómo se utilizan las siguientes funciones de Java `substring()`, `Length()` y `Math.random()`.

24. Realizar un programa que complete un vector con los N primeros números de la sucesión de Fibonacci. Recordar que la sucesión de Fibonacci es la sucesión de los siguientes números:

1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Donde cada uno de los números se calcula sumando los dos anteriores a él. Por ejemplo:

La sucesión del número 2 se calcula sumando (1+1)

Análogamente, la sucesión del número 3 es (1+2),

Y la del 5 es (2+3),

Y así sucesivamente...

La sucesión de Fibonacci se puede formalizar de acuerdo a la siguiente fórmula:

$$\text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2) \text{ para todo } n > 1$$

$$\text{Fibonacci}(n) = 1 \text{ para todo } n \leq 1$$

Por lo tanto, si queremos calcular el término "n" debemos escribir una función que reciba como parámetro el valor de "n" y que calcule la serie hasta llegar a ese valor.

Para conocer más acerca de la serie de Fibonacci consultar el siguiente link:

**<https://quantdare.com/numeros-de-fibonacci/>**

