

Tema 3: Abstracción procedimental y de datos. (ejercicios)

1. construir un programa que reciba una frase de texto por línea de órdenes y devuelva la frecuencia absoluta de aparición de las diferentes parejas de caracteres consecutivas. Visualizar por pantalla las frecuencias no nulas junto con su pareja de caracteres asociada, a razón de 6 por línea.
2. Construir un programa que extraiga todas las apariciones de un determinado carácter en una palabra almacenada en un String y devuelva por pantalla el String resultante. Tanto la palabra como el carácter a eliminar serán introducidos en el programa por línea de comandos.
3. Escribir un método que dada una cadena de caracteres, calcule la suma de todos los dígitos que hay en ella. Escribir un programa que dada una cadena de caracteres por línea de órdenes invoque al método anterior e imprima la suma de los dígitos de la cadena. Use el método printf para imprimir.
4. Escribir un método que reciba como parámetro una cadena y que devuelva una nueva cadena que corresponda a la cadena inicial invertida. Escribir un programa que imprima la cadena inicial y la nueva en el método main. La cadena inicial debe leerse por línea de órdenes. Como ejemplo para probar el programa puede usar la cadena inicial "casa", donde la salida debe ser "asac".
5. Diseñar e implementar en Java un programa que indique si una palabra es un palíndromo. Para ello use un método llamado palíndromo. Escribir un programa principal que primero compruebe que la cadena que se le pase al método palíndromo tiene menos de 15 caracteres. En caso de que la cadena sea mayor, el programa principal indicará que no se va a comprobar si la cadena es palíndroma. Nota: una palabra es un palíndromo cuando se lee igual de izquierda a derecha que de derecha a izquierda.
6. Escribir un programa que contenga cuatro métodos, aparte del main.
 - El primero de ellos, llamado suma, debe recibir dos números enteros y devolver la suma de esos dos números.
 - Un segundo método, también llamado suma, recibirá un número como parámetro y lo sumará a un entero que pedirá por teclado usando la clase Scanner. Dicho método devolverá la suma de los dos números.
 - El tercer método, llamado también suma, recibirá dos cadenas y devolverá la suma del número de caracteres de las dos cadenas.
 - El cuarto método se llamará sumaCadenas, recibirá tres cadenas como parámetros y devolverá una frase que sea la unión (concatenación) de esas 3 cadenas.En ninguno de estos métodos se debe imprimir nada. En el método main se imprimirán los valores resultantes de invocar a cada uno de los métodos.
Como parámetros actuales al invocar a los métodos se usarán los siguientes valores:
 - Primer método: 3 y 5
 - Segundo método: 7
 - Tercer método programa y modular
 - Cuarto método: voy, a, aprobar.
7. Escribir un método que compruebe si un año es bisiesto o no. El método debe recibir como argumento un año y devolver verdadero o falso dependiendo de si el año es bisiesto o no. De

acuerdo al calendario Gregoriano, un año es bisiesto si es divisible por 4, excepto aquellos divisibles por 100, que para ser bisiestos deben ser también divisibles por 400. Además, imprimir un programa que, dado un año, imprima si es bisiesto usando el método hecho anteriormente.

8. Escribir un método que compruebe si un número pasado como parámetro es impar. Si lo es, el método devolverá el siguiente número impar. Si el número no es impar, el método devolverá el número impar anterior.
9. Escribir un método que evalúe el factorial de un número N. Además, escribir un método que lea por línea de órdenes el valor de N e imprima su factorial usando el método anteriormente escrito.
10. Construir un programa con dos métodos que usen el mismo identificador (nombre).
El primer método determinará el máximo de dos números pasados como parámetro.
El segundo, obtendrá el máximo de tres números pasados como parámetros.
En ambos casos, los valores numéricos se introducirán por línea de órdenes.

11. Construir un programa que escriba el triángulo de Pascal con un número determinado de filas. dicho número se introducirá por teclado. A tal efecto, se utilizará la función combinatoria $c(n, k)$ definida como

$$c(n, k) = \frac{n!}{k!(n-k)!}$$

donde $k \leq n$.

12. Construir un programa que calcule los valores del exponente, e^x , el coseno, $\cos(x)$, y el seno, $\sin(x)$, a partir de las series siguientes:

$$e^x = \sum_{i=0}^n \frac{x^i}{i!}$$
$$\cos(x) = \sum_{i=0}^n -1^i * \frac{x^{2i}}{(2i)!}$$
$$\sin(x) = \sum_{i=0}^n -1^i * \frac{x^{2i+1}}{(2i+1)!}$$

El número de términos de la serie será el necesario para que la diferencia absoluta entre dos valores sucesivos sea menor de 10^{-3} . Úse un método para cada caso, imprimiendo los distintos resultados en el método principal.

13. Escribir un programa en Java que permite generar números aleatorios, para simular el lanzamiento de un dado. Para la generación de números aleatorios utilice la clase Random que provee Java.

Este ejemplo se realiza usando números aleatorios (pseudoaleatorios). En Java existe una clase para esta tarea, es la clase Random (aleatorio). La clase Random está definida en el paquete java.util e implementa un generador de números pseudoaleatorios. Los números aleatorios se usan mucho en juegos y en simulación. El generador de números aleatorios selecciona un valor entre un intervalo de valores. Por ejemplo, tenemos varios métodos de utilidad como son nextInt() que devuelve un entero (int) entre el intervalo de valores del tipo int. Tenemos también un nextFloat() y un nextDouble() que devuelven un real de tipo float o double seleccionado entre el intervalo 0,0 – 1,0. Para simular la obtención de un valor entero de una serie de valores se puede usar el nextInt(). Lo que hay que hacer es ‘escalar’ (cambiar la escala) del intervalo de valores del tipo int al intervalo que nos interesa. Esto se puede hacer con el operador % dividiendo por el número de valores que se desean. Como al dividir por n, los restos posibles son 0, 1, ..., n-1, ya tenemos la escala. Si en lugar de 0 a n – 1 queremos que vaya de 1 a n, sumamos 1.

Expresión	Intervalo
<code>Math.abs(nombreObjeto.nextInt()) %6+1</code>	1 a 6
<code>Math.abs(nombreObjeto.nextInt()) %10+1</code>	1 a 10
<code>Math.abs(nombreObjeto.nextInt()) %101</code>	0 a 100
<code>Math.abs(nombreObjeto.nextInt()) %11+20</code>	20 a 30
<code>Math.abs(nombreObjeto.nextInt()) %11-5</code>	-5 a 5

14. Un número se denomina mágico cuando es divisible entre 3 y 5 y no es divisible entre 10. Escribir un método que reciba como parámetros dos números e imprima los números mágicos comprendidos entre esos dos números.
15. Sean a y b dos números enteros positivos o cero. Escribir un método que calcule la potencia a^b sin usar el método pow de Java. Escribir un método main que, dados dos números a y b leídos por línea de órdenes, calcule la potencia a^b usando el método escrito anteriormente. Sólo se calculará la potencia si tanto a como b son múltiplos de 2.
16. Construir un método que devuelva el producto de los n primeros números enteros pares a partir de un número x dado. Tanto x como n se pasarán como parámetros al método. Suponer n mayor o igual a 1.
17. Realizar un programa que calcule el perímetro y el área de un círculo. Después de mostrar el resultado, el programa debe preguntar si se quiere calcular el perímetro y área de otro círculo.

18. Un club de baloncesto saca a la venta las entradas para el próximo partido. El precio de las entradas varía dependiendo de la zona del pabellón que el espectador quiere ocupar.
- El precio de una entrada para la zona de los fondos es de 5 euros. Si un aficionado adquiere más de 7 entradas para esta zona, obtiene un descuento total del 6%.
 - El precio de una entrada para la zona central es de 12 euros. Si un aficionado adquiere más de 5 entradas para esta zona obtiene un descuento total del 5%.
 - El precio de una entrada para la zona VIP es de 20 euros. Si un aficionado adquiere más de 9 entradas para esta zona obtiene un descuento total del 4%.

Diseñar una aplicación (usando las características de la programación estructurada y modular) que muestre un menú con las tres zonas del pabellón. El usuario deberá escoger la zona del pabellón para la que desea adquirir entradas. A continuación, la aplicación pedirá al usuario que introduzca el número de entradas que quiere adquirir. Posteriormente, el programa mostrará por pantalla el importe en euros que el aficionado debe abonar. Utilice la clase Scanner para leer los datos por teclado y el método printf para mostrar la salida del programa por pantalla.

19. Diseñar un programa que simule el funcionamiento de una sencilla calculadora con un conjunto básico de operaciones. El programa consistirá básicamente, en la presentación de un menú al usuario en el que se le indicarán las operaciones de las que dispone. Estas serán: suma, resta, multiplicación, división y potencia de 2 operandos, que deben ser números reales. Además, existirá una opción dentro del menú para finalizar el programa. Si la opción que introduce el usuario no es la de finalización, el programa le solicitará dos datos necesarios para realizar la operación elegida. El programa debe controlar la introducción errónea de datos, mostrando un mensaje de error cuando esto se produzca, y ofreciendo al usuario la posibilidad de que vuelva a introducir los datos correctamente.
20. Construir un programa que obtenga el término n de la serie de Fibonacci. El valor de n deberá leerse por teclado. La serie de Fibonacci es una secuencia de enteros positivos, cada uno de los cuales es la suma de los dos anteriores. Los dos primeros números de la secuencia son 0 y 1. La serie se define como:

$$\begin{aligned} \text{Fibonacci}(n) &= n, & \text{si } n \leq 1 \\ \text{Fibonacci}(n) &= \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2), & \text{si } n > 1 \end{aligned}$$

El programa tendrá, además del método main, un método iterativo para calcular el término de la serie. El método main deberá llamar al método e imprimir, usando printf, el término n de la serie. Si el usuario introduce un valor negativo, el programa debe emitir un aviso.

21. Escribir un programa que, dados dos enteros introducidos por teclado, evalúe su máximo común divisor usando el algoritmo de Euclides implementado en un método. El algoritmo tal y como lo propuso Euclides es el siguiente:
- a) Tómesese el resto del cociente m / n
 - b) Si el resto es cero, entonces n es el máximo común divisor
 - c) Si el resto es distinto de cero, se hace $m = n$ y $n = \text{resto}$
 - d) Se vuelve al punto primero

22. Dos números son amigos si cada uno de ellos es igual a la suma de los divisores del otro. Por ejemplo, 220 y 284 son amigos, ya que:

- Suma de divisores de 284: $1 + 2 + 4 + 71 + 142 = 220$
- Suma de divisores de 220: $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$

Construir un método que determine si dos números dados como parámetros son amigos o no. A continuación, realizar un programa que muestre todas las parejas de números amigos menores o iguales que n, siendo n un número introducido por teclado. El programa debe usar el método previamente definido.

23. Los factores primos de un número entero son los números primos divisores exactos de ese número entero cuyo producto es igual al número original. Los factores primos no incluyen el 1 pero sí cada copia de cada número primo. Por ejemplo, los factores primos de 90 son 2, 3, 3, 5; los factores primos de 1092 son 2, 2, 3, 7, y 13. Para encontrar los factores primos de un número el procedimiento a seguir es:

- Comprobar si el número es divisible por el menor número primo posible (considerar 2). Si es divisible, este número primo es el primer factor primo.
- Dividir el número por el primer factor primo y comprobar si el resultado es divisible por dicho factor. Si es así, tendríamos el segundo factor primo (que sería repetido). Esta operación se repite hasta que el resultado no pueda dividirse por ese factor primo.
- Cuando el resultado no pueda volverse a dividir por ese número, buscar el siguiente número primo posible para continuar dividiendo, repitiendo los pasos anteriores. Desde el punto de vista algorítmico es suficiente con ir probando los siguientes números enteros.
- Cuando el cociente entre el resultado y el factor primo sea 1, acabar.

Construya un método que reciba como argumento un número y devuelva la suma de los factores primos de ese número, imprimiéndolos en el método main.

24. Escribir un programa que cree una matriz con los caracteres de la 'a' a la 'j' y a continuación sustituya todas las vocales que se encuentren en la matriz por el carácter @

25. Construir un programa que obtenga la matriz suma de dos matrices: A (con dimensiones M y N) y B (con dimensiones P y Q)

26. Realizar un programa que, dada una secuencia de enteros separados por espacios en blanco e introducidos por línea de órdenes, visualice por pantalla la subsecuencia creciente de mayor tamaño. Si hay varias secuencias del mismo tamaño, se tomará la situada más a la izquierda.

27. Escribir un método que reciba como parámetro un vector de enteros A y devuelva un nuevo vector B que sea igual al vector A, pero desplazado hacia la derecha, es decir, todos sus elementos, han sido desplazados una posición hacia la derecha. Ejemplo:

$$A = \{1, 2, 3, 4\} \rightarrow B = \{4, 1, 2, 3\}$$

Nota: Téngase en cuenta que el último elemento del vector A, al ser desplazado hacia la derecha, pasa a ser el primer elemento de B.

28. Escribir un método que reciba dos vectores A y B de enteros del mismo tamaño y devuelva el sumatorio de la multiplicación inversa de ambas matrices, es decir, multiplique el primer elemento de A por el último de B, el segundo elemento de A por el penúltimo de B, y así sucesivamente, sumándolos todos.

Ejemplo:

$$A = \{1, 2, 3\}, \quad B = \{6, 7, 8\}.$$

El resultado sería: $1 \cdot 8 + 2 \cdot 7 + 3 \cdot 6 = 40$

29. Escribir un método que acepte una matriz monodimensional de enteros y devuelva la suma de los valores almacenados en dicha matriz.
30. Escribir un método que acepte una matriz monodimensional de enteros e imprima el número mayor, el menor y el número de elementos de la matriz.
31. Construir un programa que cree dos matrices bidimensionales, una de tipo String y otra de tipo int. Cada matriz debe ser de tamaño 3 x 2. El programa debe leer por teclado los elementos de ambas matrices. Después, deberá imprimir ambas matrices. A continuación deberá sumar un número, que se introducirá por teclado, a cada elemento de la matriz de enteros y concatenar una palabra, que también se introducirá por teclado, a los elementos de la matriz cadena. Finalmente, se volverá a imprimir cada matriz. deberá utilizarse un método para cada una de las tareas que realiza el programa.
32. Escribir un método que construya y devuelva matrices cuadradas de cualquier dimensión mayor o igual a 1, cuyos elementos sigan el patrón mostrado en las matrices de la figura (cada elemento es la suma de sus índices). La dimensión se pasará como parámetro al método El resultado se imprimirá en el método main.
33. Escribir un método que reciba como parámetro una matriz bidimensional con valores numéricos enteros y devuelva una matriz unidimensional de longitud 5 que contenga los 5 primeros elementos pares de la matriz bidimensional., recorriéndola por filas. Si hay menos de 5 elementos pares en la matriz bidimensional, rellene las posiciones restantes de la matriz unidimensional con el valor -1. Además, escribir un programa que declare e inicialice una matriz bidimensional y, usando el método anterior, imprima por pantalla la matriz resultante de ejecutar dicho método.
34. Construir un método que reciba una matriz bidimensional de enteros y devuelva un vector de 5 posiciones donde se guarden los 5 primeros elementos pares de una matriz bidimensional recorriéndola de abajo a arriba y de derecha a izquierda. Si hay menos de 5 elementos pares en la matriz, rellenar las posiciones restantes del vector con el valor -1.

35. Construir en Java dos métodos que reciban como único parámetro una matriz de enteros. El primer método debe devolver la media aritmética de los enteros de la matriz. El segundo método debe devolver el valor máximo de la matriz. Construya un programa principal que lea una matriz de enteros, invoque a ambos métodos e imprima los valores devueltos.
36. Construir un programa que declare 2 matrices unidimensionales, una en la que se almacenen nombres de alumnos y otra que contenga sus notas. Ambas matrices serán de 10 elementos. El propio programa debe inicializar las matrices de nombres y de notas. El programa debe imprimir la lista de nombres con su nota al lado. Después, se deberá introducir en una nueva matriz llamada `alumnosAprobados` los alumnos que hayan aprobado, e imprimir esta matriz.
37. Construir un programa que muestre por pantalla el resultado de multiplicar dos matrices bidimensionales de números enteros: La matriz A (con dimensiones M y N) y la matriz B (con dimensiones O y P). Para implementar la entrada de datos, diseñe un método que solicite al usuario introducir las dimensiones y los valores de cada matriz.
38. Implementar un programa principal que cree una matriz bidimensional con los siguientes valores: $\{\{1, 2\}, \{3, 4\}\}$. Además, implementar un método que intercambie el primer valor de la matriz con el último, por lo que la matriz quedaría así:
- $$\begin{pmatrix} 4 & 2 \\ 3 & 1 \end{pmatrix}$$
- El método debe devolver la matriz. Imprimir la matriz (en forma matricial) antes y después de realizar el intercambio de valores, invocando un método que habrá que implementar.
39. La matriz traspuesta de una dada se define como aquella que intercambia filas por columnas (el elemento (i, j) pasa a ser el (j, i)). Escribir un método que transponga una matriz cuadrada. El método debe devolver la matriz traspuesta sin sobrescribir la matriz original. Para probar el programa, construir un método `main` que inicialice una matriz y calcule su traspuesta imprimiéndola posteriormente.
40. Escribir un programa que, dada una matriz que contenga los diez primeros enteros mayores que cero, permita sustituir un elemento de una determinada posición de la matriz por otro introducido por teclado. Esta operación se debe realizar usando un método. Construir otro método que devuelva la suma de todos los elementos de la matriz modificada. En el método `main`, imprimir la matriz modificada y su suma. El programa debe indicar si se ha introducido una posición de la matriz no válida.
41. Escribir un programa que calcule el máximo y el mínimo de n números introducidos como argumentos por la línea de órdenes.

42. Escribir un programa que genere un boleto de lotería primitiva con el número de apuestas elegido por el usuario. El programa hará uso del concepto de modularización, conteniendo los métodos necesarios para la generación del boleto. En el método main se leerá el número de apuestas y se imprimirá el boleto (números de cada apuesta, reintegro y precio) Un boleto de lotería primitiva se compone de n apuestas de 6 números enteros elegidos al azar en el intervalo de 1 a 49 y un reintegro elegido al azar entre 0 y 9. El coste de cada apuesta es de 1 euro.
43. Escribir un método que reciba como parámetro una matriz bidimensional de cadenas. El método debe devolver una matriz de caracteres, donde en las columnas pares se colocará el primer carácter de la cadena correspondiente y en las columnas impares el último carácter de la palabra correspondiente. Construir un programa de prueba.
44. Escribir un método que reciba como parámetro una matriz A que sea cuadrada y real. En dicho método debe construirse una matriz B, también real, que guarde en las filas pares la parte entera de las filas pares de la matriz A, y en las impares incremente en 5 unidades el valor de las filas impares de la matriz A. Al finalizar el método, la matriz que se pasó como parámetro debe conservar sus valores originales. El método deberá devolver la matriz B.
45. Escribir un método que invierta un vector de enteros. El método debe recibir como argumento el vector a invertir y devolver, en otro vector distinto, el vector invertido
46. Dadas dos matrices reales A y B, de dimensiones $n \times n$, es decir, dos matrices cuadradas, escribir un método en Java que acepte las dos matrices como parámetros y devuelva la matriz suma de estas dos.
47. La criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores que un número natural dado N. Para ello, se forma una tabla con todos los números naturales comprendidos entre 2 y N y se van eliminando los números que no son primos de la siguiente manera: cuando se encuentra un número entero que no ha sido eliminado, ese número se considera primo y se procede a eliminar todos sus múltiplos. El proceso termina cuando el cuadrado del mayor número primo encontrado es mayor que N. Escribir un programa en Java que lea un entero N y escriba los números primos existentes entre 1 y N. El programa debe implementar el cálculo de los primos en un método que aplique la criba de Eratóstenes.