

Ejercicios de Metodología de la programación: Algoritmos y programas

1.- Analizar el algoritmo siguiente que permite pelar un número suficiente de patatas que se hallan en un cesto. Debe tenerse en cuenta que el cesto puede estar vacío en un momento dado:

```
Si cesto vacío
    ir a llenar el cesto
Finsi
Pelar 1 patata
Mientras núm de patatas insuficiente
    Si cesto no vacío
        Pelar 1 patata
    Finsi
    Si cesto no vacío
        Pelar 1 patata
    Sino
        ir a llenar el cesto
        Pelar 1 patata
    Finsi
Finmientras
```

Estudiar como "pelar un número impar de patatas teniendo en cuenta el estado del cesto". Verifíquese en función del número de veces que el cesto es vaciado y vuelto a llenar. Encontrar ejemplos de valores que confirmen los distintos casos hallados.

Ejercicios de Metodología de la programación: Objetos de un programa y expresiones

2.- Dadas las siguientes variables y constantes

1	4	10	3,141592	2,718281
--	--	--	-----	-----
X	Y	Z	PI	E

Evaluar las expresiones:

- a) $2 * X + 0,5 * Y - 1 / 5 * Z$
- b) $PI * X^2 > Y \text{ OR } 2 * PI * X \leq Z$
- c) $E^{(X - 1) / (X * Z) / (X / Z)}$
- d) $"DON " + "JUAN" = "DON JUAN" \text{ OR } "A" = "a"$
- e) $"12" + "12" = "24"$
- f) $"12" + "12" = "1212"$
- g) $"hola" = "hola"$
- h) $"hola" = "hola "$
- i) $12 + 12 = 24$
- j) $\text{Verdad AND Verdad} = \text{Falso}$
- k) $\text{NOT Falso} = \text{Verdad}$
- l) $(\text{Verdad AND Verdad}) \text{ OR Falso} = \text{Verdad}$
- ll) $(\text{Falso OR Falso}) \text{ AND Falso} = \text{Verdad}$
- m) $\text{NOT (Verdad AND Falso)} = \text{Falso}$

Ejercicios de Metodología de la Programación: Estructura general de un programa. Notación Pseudocodificada de programas

3.- Programa que lee un número que corresponde al radio de una circunferencia y calcula e imprime la longitud de la misma y el área del círculo correspondiente

4.- Programa que lee dos números y calcula e imprime su suma, resta, producto y división

5.- Introducir un número por teclado. Que nos diga si es positivo o negativo

6.- Introducir un número por teclado. Que nos diga si es par o impar

7.- Algoritmo que calcule el máximo de dos números introducidos por teclado

8.- Algoritmo que calcule el máximo de 3 números introducidos por teclado

9.- Programa que lee los nombres de dos personas en formato "1^{er} apellido-blanco-2ºapellido-blanco-nombre" y los imprime en orden alfabético

10.- Diseñar un algoritmo para leer tres valores reales, a, b y c, correspondientes a los lados de un triángulo, ordenados de mayor a menor, y determinar el tipo de triángulo de acuerdo con las condiciones:

$a \geq b + c$ no forman triángulo

$a^2 = b^2 + c^2$ rectángulo

$a^2 > b^2 + c^2$ obtusángulo

$a^2 < b^2 + c^2$ acutángulo

Comprobar al principio del algoritmo el orden requerido

11.- Introducir una cantidad por teclado. Si es menor de 500, sumarle el 50 por 100; si es mayor o igual a 500 pero menor de 1000, sumarle el 7 por 100; si es mayor o igual a 1000 y menor o igual a 5000, sumarle el 15 por 100, y si es mayor de 5000, restarle el 5 por 100

12.- Programa que lee una calificación entre 0 y 10 y la transforma en nota alfabética según la tabla:

Nota numérica	Nota Alfabética
-----	-----
$0 \leq \text{NOTA} < 3$	M.D.
$3 \leq \text{NOTA} < 5$	INS.
$5 \leq \text{NOTA} < 6$	SUF.
$6 \leq \text{NOTA} < 7$	BIEN
$7 \leq \text{NOTA} < 9$	NOT.
$9 \leq \text{NOTA} \leq 10$	SOBR.

13.- Programa que lee 3 números y los imprima ordenados de mayor a menor, indicando si fueron introducidos en ese orden

14.- Crear un programa que nos escriba el día completo de la semana si se introducen por teclado los dos primeros caracteres

15.- Programa que lea un número correspondiente a un año y escriba si es o no bisiesto. (Son bisiestos los múltiplos de 4, excepto los múltiplos de 100 que no lo son de 400)

16.- Introducir 5 números por teclado. Decir si esos números son mayores o menores de 10

17.- Hacer un pseudocódigo que imprima los números pares entre 0 y 100

18.- Hacer un programa que imprima y sume los 100 primeros números

19.- Hacer un pseudocódigo que imprima los números impares hasta el 100 y que imprima cuantos impares hay

20.- Hacer un programa que calcule independientemente la suma de los pares y los impares de los números entre 1 y 1000, utilizando un switch

21.- Hacer un pseudocódigo que imprima todos los números naturales que hay desde la unidad hasta un número que introducimos por teclado

22.- Programa que lea 100 números y cuente cuántos de ellos son positivos

23.- Programa que calcule e imprima la suma y el producto de los números pares comprendidos entre el 10 y el 30 (ambos inclusive)

24.- Algoritmo que lea una secuencia de 100 números y obtenga e imprima cuántos hay positivos, negativos y nulos.

25.- Algoritmo que lea una secuencia de números no nulos, terminada con la introducción de un 0, y obtenga e imprima el mayor y la suma, imprimiendo un mensaje de si se ha leído algún número negativo

26.- Algoritmo que calcule la media de 5 números introducidos por teclado

27.- Hacer un pseudocódigo que sólo nos permita introducir S o N

28.- Algoritmo que calcule la media de n números introducidos por teclado

29.- Algoritmo que calcule la media de una serie de números introducidos por teclado, finalizando cuando el usuario responda "N" a la pregunta de si quiere introducir más datos

30.- Algoritmo que lea una serie de números enteros positivos, finalizando con un número negativo, y calcule su media (sin contar el negativo)

31.- Hacer un pseudocódigo que imprima los números del 100 al 0, en orden decreciente

32.- Programa que imprima la tabla de multiplicar de un número entero positivo

33.- Hacer un pseudocódigo que imprima el mayor y el menor de una serie de cinco números introducidos por teclado

34.- Introducir tantos números como queramos. Decir si son mayores o menores que un número que previamente introduciremos por el teclado

35.- Introducir una serie de nombres de personas con su sueldo. Imprimir el nombre y el sueldo de la persona que más gana y de la que menos. Si hay varias con igual sueldo, imprimir la primera de todas. Contar el número de personas introducidas

36.- Imprimir 10 veces, de una forma alternativa, Hola y Adiós

37.- Introducir dos números por teclado. Imprimir los números naturales que hay entre ambos números empezando por el más pequeño, contar cuántos hay y cuántos de ellos son pares. Calcular la suma de los impares

38.- Hacer un programa que imprima y cuenta los múltiplos de 3 que hay entre 1 y 100

39.- Programa que obtenga e imprima la lista de los divisores de un número N entero positivo

40.- Programa que calcule e imprima el interés producido por un capital a partir del Capital, Rédito, Tiempo de imposición

41.- Introducir un número por teclado menor de 500. Imprimir, sumar, sumar los cuadrados y contar los números que están separados entre sí 8 posiciones desde ese número al 500

42.- Calcular la suma de los cuadrados de los números pares entre 0 y 100

43.- Imprimir y contar los números que son múltiplos de 2 o de 3 que hay entre 1 y 100

44.- Hacer un programa que determine cuántas cifras posee un número entero positivo introducido por teclado.

45.- Imprimir, contar y sumar los múltiplos de 2 hasta un número que introducimos por teclado

46.- Imprimir los múltiplos de 3 que hay entre dos números determinados de una forma alternativa, es decir uno sí y otro no

47.- Introducir 2 números inferiores a 50 por teclado. Al más pequeño aumentarlo de 5 en 5 y al mayor disminuirlo de 2 en 2. Imprimir ambas series de números, de una forma alternativa hasta que el menor supere al mayor

48.- Hallar el factorial de un número

49.- Calcular e imprimir los números perfectos menores que un número introducido por teclado. Un número es perfecto si la suma de sus divisores, excepto él mismo, es igual al propio número

50.- Calcular el menor número primo mayor que un número introducido por teclado

51.- Algoritmo que lea un número X y otro positivo N y calcule la N-ésima potencia de X

52.- Algoritmo que obtenga el producto de dos números enteros positivos mediante sumas sucesivas

53.- Obtener el cociente y resto de dos números enteros positivos mediante restas

54.- Se dispone de monedas de 1, 5, 25, 50 y 100 pesetas. Realizar un algoritmo que nos diga la descomposición mínima en monedas de una cantidad introducida por teclado

55.- "JUEGO DE MARIEBAND": Este sencillo juego presenta la particularidad de ofrecer una estrategia ganadora a quien juegue en segundo lugar. Las reglas del juego son las siguientes:

Se dispone inicialmente de un conjunto de 21 cerillas, de las cuales cada uno de los dos jugadores podrá tomar en su turno entre 1 y 4 cerillas. Quien tome la última pierde.

La estrategia del jugador que juega en segundo lugar es completar a 5 el número de cerillas tomadas anteriormente por el primer jugador. Así, el primer jugador se encontrará delante de un conjunto que contendrá sucesivamente 21, 16, 11, 6 y 1 cerillas y será él quien tendrá que tomar la última.

Construir un algoritmo que otorgue al computador el papel del segundo jugador y le haga aplicar la estrategia ganadora. El problema deberá detectar las trampas del primer jugador (comprobar que elige entre 1 y 4 cerillas)

56.- El término general de la sucesión de Fibonacci viene dado por:

$$F_1 = 1; \quad F_2 = 1; \quad F_i = F_{i-1} + F_{i-2}$$

Escribir un algoritmo para la obtención de n elementos de la sucesión

57.- Modificar el algoritmo anterior de tal forma que al introducir un número nos indique si pertenece o no a la serie de Fibonacci

58.- Resolver una ecuación de segundo grado de la forma: $ax^2 + bx + c = 0$

El algoritmo debe leer los valores de a, b y c, escribiendo los valores de x que correspondan a las soluciones reales, emitiendo un mensaje para las soluciones complejas

59.- Algoritmo que calcule e imprima el valor del número e como suma de la serie:

$$\sum 1/i! = 1/0! + 1/1! + 1/2! + \dots + 1/n!$$

El resultado se obtendrá con una precisión superior a 10^{-3} , la cual será el dato de entrada al programa; éste ha de obtener y acumular sumandos de la serie hasta el primero que sea inferior a la precisión introducida.

Modificar el programa para que nos muestre además el número de iteraciones que ha tardado

60.- Escribir un programa que calcule la raíz cuadrada de un valor A mediante la sucesión X_0, X_1, \dots donde:

$$X_0 = A; \quad X_{(i+1)} = (X_i + A/X_i)/2$$

Acabar cuando $X^2 - A$ sea menor que un valor épsilon = 0,0001

61.- Escribir un algoritmo que imprima los 100 primeros números que sean primos. Un método para encontrar los números primos consiste en buscar a partir de 3 entre los números impares aquellos que no acepten otros divisores que ellos mismos y la cifra 1.

Otro método más elegante consiste en que a partir de 5 incluido, todo número primo es de la forma $6n \pm 1$ siendo n entero. (si es primo cumple la fórmula anterior pero si cumple la fórmula anterior puede no ser primo, por ejemplo para $n = 4$ 23 es primo pero 25 no)

62.- Programa que calcule si dos números enteros positivos son amigos. (Dos números son amigos si la suma de los divisores del primero excepto él mismo es igual al segundo y viceversa)

63.- Encontrar todos los números enteros a y b comprendidos entre 1 y 100 de manera que $a^2 + b^2 = c^2$ (cuadrado perfecto). Se quiere hacer notar de entrada que será necesario eliminar dos veces las soluciones idénticas por permutación. Por ejemplo:

$a=3 \quad b=4 \quad c=5$ y $a=4 \quad b=3 \quad c=5$ son soluciones idénticas.

Para evitar esto se buscarán números b que sean mayores que a

64.- Se denominan números de Armstrong aquellos números que son iguales a la suma de los cubos de las cifras que los integran. Por ejemplo 153 es un número de Armstrong, ya que $153 = 1^3 + 5^3 + 3^3$.

Escribir un algoritmo que nos proporcione todos los números de Armstrong comprendidos entre 1 y 2000

65.- Un número perfecto es aquél que es igual a la suma de todos sus factores incluyendo el 1 y excluyéndose el mismo. Así 6 es un número perfecto porque sus factores son 6, 3, 2, 1 y la suma de 3, 2, 1 es 6. Contrariamente 12 no es un número perfecto, porque sus factores son 12, 6, 4, 3, 2, 1 y la suma de 6, 4, 3, 2, 1 es distinta de 12.

Escribir un algoritmo que permita encontrar todos los números perfectos que sean menores que un número N leído por teclado