

Ejercicios del Tema 2: Diseño de Algoritmos Simples

En los siguientes ejercicios se pretende descubrir un algoritmo para solucionar distintos problemas, de forma que sólo se empleen instrucciones elementales: Operaciones aritméticas (suma, resta, multiplicación...), asignación, entrada y salida, bucles, selección...

Para cada ejercicio intente examinar cómo se efectúa la operación '*a mano*'. Se trata de ver que cuando nosotros solucionamos un ejercicio también lo solucionamos ejecutando distintas operaciones una a una. Despues, trate de descubrir las operaciones repetitivas, que serán los bucles, y las operaciones en las que pregunta por una condición (Si se cumple que... entonces... y si no...), que serán operaciones de selección. Por último, construya el algoritmo siguiendo una forma de representación de algoritmos.

1. Diseñar un algoritmo que lea 3 números enteros positivos distintos e imprima el mayor valor de los 3. Supondremos que los 3 números que se leen son distintos. Para este algoritmo sólo es necesario usar estructuras de selección y no de iteración (no son necesarios los bucles).
2. Efectuar un algoritmo que averigue si un número positivo es par o impar. El programa leerá un número entero positivo y dará el resultado. Supondremos que tenemos un operador para el módulo MOD, es decir, el resto de una división entera. Intente hacerlo también suponiendo que no tenemos el operador MOD.
3. Diseñar un algoritmo para sumar los 100 primeros números naturales. El algoritmo no leerá ningún valor. Simplemente mostrará el resultado.
4. Diseñar un algoritmo para sumar los 100 números siguientes a un valor entero positivo que deberá leer. El algoritmo primero leerá un valor entero positivo, luego calculará el dato y por último mostrará el resultado.
5. Escribir un algoritmo para hallar el valor del MCD (Máximo Común Divisor) de 2 números enteros positivos que deberán leerse al principio del algoritmo.
6. Escribir un algoritmo para hallar el valor del MCM (Mínimo Común Múltiplo) de 2 números enteros positivos que deberán leerse al principio del algoritmo.
7. Calcular el factorial de un número n ($n!$), teniendo en cuenta que: $0! = 1$ y que, en general tenemos que: $n! = n * (n-1) * (n-2) * \dots * 3 * 2$. El factorial no está definido para números negativos.
8. Diseñar un algoritmo que lea 3 números enteros positivos distintos e imprima el mayor y el menor valor de los tres. Observe que hay muchas formas de hacerlo. Intente encontrar la más eficiente (la que realice menos operaciones entre los números).
9. Diseñar un algoritmo para leer una lista de números positivos y escribir, después de leerla, el valor máximo y el mínimo de esa lista. El algoritmo leerá un número tras otro hasta que se introduzca un número negativo. Es decir, el algoritmo dejará de leer números cuando lea un número negativo y, entonces, mostrará los valores resultantes.

10. Hacer un algoritmo que calcule la suma de los 100 números siguientes al MCD de dos números dados. Este ejercicio es una mezcla de los ejercicios 4 y 5.

11. Diseñar un algoritmo para convertir temperaturas en grados Farenheit a grados Celsius. El programa terminará cuando lea una temperatura igual a 999. Como todo el mundo sabe, la relación entre grados Farenheit (F) y grados Celsius (C) viene dada por: $C = 5/9(F - 32)$.

12. Diseñar un algoritmo que lea 2 puntos en el plano (4 números reales): (X1, Y1) y (X2, Y2) y devuelva la distancia euclídea entre ellos. Para ello se usará el teorema de Pitágoras.

13. Diseñar un algoritmo que lea un punto en el plano (2 números reales): (X, Y), y calcule el cuadrante al cual pertenece el punto (1, 2, 3 ó 4). Mostrará un valor 0 si el punto está en un eje de coordenadas.

14. Diseñar un algoritmo que lea un texto carácter a carácter y reemplace cada cadena de uno o más espacios en blanco por un sólo espacio. Es decir, en la salida no se deben escribir más de un espacio seguido.

15. Suponga que la CEE y la comunidad internacional se dan cuenta, por fin, de que el Mediterráneo necesita medidas urgentes para su recuperación, ya que se está muriendo por los vertidos incontrolados de fábricas y de actividades turísticas, además de la pesca incontrolada, pesca de inmaduros (pezqueñas) y el empleo de las peligrosas redes de deriva. Así, se decide poner un límite en los Kg. de pesca que puede recoger en un mes un barco pesquero. Desarrolle un algoritmo al que primero se le dé el límite máximo permitido por la ley y luego se le vayan dando los pesos de lo que se va pescando. Cada vez que se introduzca un nuevo peso se mostrará el total de Kg. pescados hasta ese momento. Cuando se supere el máximo permitido se debe dar la alarma y terminar el algoritmo, indicando el total de Kg. pescados y el número de Kg. en los que se ha sobrepasado el límite legal. El algoritmo también terminará si introducimos un 0 como Kg. pescados y deberá indicar un error si se introduce un número negativo.

16. Diseñar un algoritmo para calcular el n-ésimo término de la sucesión de Fibonacci, donde n es un entero positivo, teniendo en cuenta que dicha sucesión está definida por:

$$\begin{aligned} \text{Fibonacci } (0) &= 0 \\ \text{Fibonacci } (1) &= 1 \\ \text{Fibonacci } (n) &= \text{Fibonacci } (n-1) + \text{Fibonacci } (n-2) \end{aligned}$$

17. Hacer un programa que pida:

- Total de kilómetros recorridos.
- Precio de la gasolina por litro.
- Dinero de gasolina invertido en el viaje.
- Tiempo que se ha tardado en el viaje (en horas y minutos).

y calcule los siguientes datos:

- Consumo de gasolina (en litros y pesetas) por cada 100 kilómetros.
- Consumo de gasolina (en litros y pesetas) por cada kilómetro.
- Velocidad media (en Km/h y en m/sg).