

Boletín de Problemas I (Parte 1). Curso 2010-2011

DISEÑAR ALGORITMOS PARA:

1. Sumar dos números reales que constituyen la entrada del algoritmo.
2. Determinar si dos números enteros, que constituyen la entrada del algoritmo, son par.
3. Calcular las raíces reales de una ecuación de segundo grado ($ax^2 + bx + c = 0$).

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

4. Obtener la suma de una lista de n números enteros que constituyen la entrada del algoritmo.
5. Contar la cantidad de valores que tiene una lista de caracteres que constituyen la entrada del algoritmo. La lista finaliza cuando se lee el carácter 'Z'. Nota: El carácter leído no se utiliza para nada.
6. Hallar el máximo entero positivo n tal que $2^n \leq N$. El valor de N es una entrada del algoritmo.
7. Escribir un algoritmo para calcular $n!$ ($n! = n * n-1 * n-2 \dots 2 * 1$) y guardar el resultado en la variable *fact*.
8. Calcular el promedio de los valores de una lista de n números enteros. Tanto el valor de n como cada uno de los valores de la lista constituyen entradas del algoritmo.
9. Calcular el promedio de una lista de números enteros positivos que constituyen la entrada del algoritmo. La lista finaliza cuando se lee el número cero o uno negativo.
10. Determinar el máximo y el mínimo valor de una lista de n números reales que constituyen la entrada del algoritmo.
11. Determinar el resultado de la expresión N^m . Tener en cuenta que el hipotético ordenador donde se va a ejecutar este algoritmo NO posee ningún operador o función que permita realizar directamente el cálculo antes citado. Por ello el algoritmo debe resolverlo mediante multiplicaciones sucesivas, es decir:

$$N^m = \prod_{i=1}^m N \quad \text{Ejemplo: } 8^5 = 8 * 8 * 8 * 8 * 8; 2^4 = 2 * 2 * 2 * 2$$

12. Calcular el máximo común divisor (mcd) de los números enteros no negativos m y n .
13. Calcular el mínimo común múltiplo (mcm) de dos números enteros no negativos m y n .
 $\text{mcd}(m,n) * \text{mcm}(m,n) = m * n$
14. Determinar la posición que tiene el número k en la lista *numeros*, que tiene n elementos y constituye la entrada del algoritmo. Los números entran de uno en uno. El resultado se queda almacenado en la variable *pos*. Si el número no se encuentra en la lista $\text{pos} = 0$.
15. Mostrar el nombre del día de la semana que le corresponde a un número entero positivo que se introduce por teclado. El número está comprendido entre el 0 y el 6, ambos inclusive. Al Domingo le corresponde el valor 0, al Lunes el 1, y así sucesivamente, hasta llegar al Sábado que le corresponde el valor 6.
16. Leer una serie de números enteros que termina al introducir un 0. Indicar la suma de todos ellos, y cuántos números de los que se han leído son positivos y cuántos negativos.
17. Escribir el equivalente en binario de un número entero que constituye la entrada del algoritmo. Se deja a elección del estudiante si se comienza a escribir por el bit más significativo o por el menos significativo.

18. Suponga que un país 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.
19. Invertir un número entero que constituye la entrada del algoritmo. Se debe suponer que los dígitos que conforman el número se almacenan en una matriz (vector) de caracteres, por lo que se puede acceder a cada uno de ellos, de forma independiente, mediante el índice correspondiente de la matriz. El número tiene una cantidad indeterminada de dígitos. Se considera que se deben almacenar dígitos en la matriz mientras entren por teclado caracteres entre '0' y '9'.
20. Contar las vocales minúsculas que contiene la matriz de caracteres *mcaracter* que posee *long* elementos. A los efectos del algoritmo se supone que ambas variables ya se encuentran en memoria con los valores actualizados.
21. Contar las palabras que contiene la matriz de caracteres *mcaracter* que posee *long* elementos. A los efectos del algoritmo se supone que ambas variables ya se encuentran en memoria con los valores actualizados y que las palabras están separadas por un espacio entre cada una.
22. Repetir el ejercicio 21 para el caso que las palabras puedan estar separadas por más de un espacio o tabulaciones. Las tabulaciones se identifican por la constante *TAB*.
23. Guardar en una matriz de caracteres el contenido concatenado de otras dos matrices del mismo tipo de dato. Las entradas del algoritmo son: el contenido de las dos matrices origen, la cantidad de caracteres (longitud) de cada una de las matrices origen y la longitud de la matriz destino. Debe tenerse en cuenta que la matriz destino debe contener las dos matrices.

Ejemplo:

origen1 = "ABCDEF"

origen2 = "ZYXVU"

El resultado para la matriz destino es "ABCDEFZYXVU"

EJERCICIOS ADICIONALES

(Ejercicios no puntuables, pero muy interesantes)

- I. Rehacer el ejercicio 17 para que, de acuerdo al valor de una entrada adicional, el número se pueda convertir a cualquiera de las tres bases (binario, octal o hexadecimal). El resultado debe ser escrito de forma que el dígito más significativo quede en la parte izquierda del resultado. Suponed que el número tiene 64 bits como máximo. Sugerencia: Utilizad una matriz para guardar el resultado.
- II. Escribir la serie: 1, -1, 2, -2, 3, -3,..., 9, -9, 10, -10, ..., *n*, -*n*. La cantidad de valores a escribir (*n*) constituye una entrada del algoritmo. Sugerencia: Utilizad una estructura PARA. Necesitaréis, además, una variable lógica que actúe de bandera: Por ejemplo, se puede llamar Signo (inicialmente vale Verdadero) e irá tomando alternativamente los valores Verdadero y Falso, que se corresponderá con el signo positivo y negativo, respectivamente.
- III. Mostrar en pantalla la serie: 0, 50, 2, 48, 4, 46, 6, 44,..., 50, 0. Sugerencia: Utilizad una estructura MIENTRAS y dos contadores.

- IV. Determinar si un número entero positivo, que constituye la entrada del algoritmo, es perfecto.

Un número es perfecto si la suma de sus divisores propios, excepto él mismo, es igual al número. Ejemplo: El número 6 es perfecto porque la suma de sus divisores propios (1, 2, 3) es igual a él mismo.

Un número N es divisor propio de M si el resto de la división entera de M/N es cero.

Sugerencia: Guardar los divisores propios del número que constituye la entrada del algoritmo en una matriz. Suponer que un número no va a tener más de 50 divisores propios.

- V. Determinar si dos números enteros positivos, que constituyen la entrada del algoritmo, son amigos.

Los números A y B son amigos si la suma de los divisores propios de A , excepto él mismo, es igual a B y viceversa. Ejemplo: El par 220, 284 son números amigos ya que la suma de los divisores propios de 220 (1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110) es 284. También la suma de los divisores propios de 284 (1, 2, 4, 71 y 142) es 220.

Sugerencia: Guardar los divisores propios de los números que constituyen la entrada del algoritmo en sendas matrices. Suponer que un número no va a tener más de 50 divisores propios.