

## Boletín de Problemas IV. Curso 2010-2011

**Nota:** En los ejercicios será necesario hacer un programa para la prueba de la función o procedimiento especificado.

1. Modificar la función Factorial, analizada en clase, para convertirla en una subrutina que permita obtener el factorial de cualquier número entero a través de uno de sus parámetros. (**No entregar: Resuelto en clase**)
2. Realizar una función que calcule la potencia de un número. La base es real y el exponente entero negativo o positivo (recuerda que  $a^{-b} = (1/a^b)$ ).
3. Repetir el ejercicio anterior, en lugar de una función, con un procedimiento *Sub* que devuelva el resultado en un tercer parámetro.
4. Implementar una función que determine si un número, que se le pasa como parámetro, es primo. Dicha función devolverá un tipo de dato *Boolean* con valor *True* si el número cumple con la condición o *False* en caso contrario. Utilizar dicha función para determinar la cantidad de números primos en un determinado rango que puede estar entre 0 y 200.000.  
Tener presente que el usuario puede equivocarse y proporcionar un valor final más pequeño que el inicial.
5. Escribir un programa que tenga una subrutina que reciba una cadena de caracteres como parámetro y devuelva en el mismo la cadena invertida. Por ejemplo “casa” se convierte en “asac”.
6. Realizar una función que reciba un parámetro de tipo cadena de caracteres y devuelva el valor lógico (*boolean*) *True* si la cadena constituye un palíndromo y *False* si no lo es.  
**Nota:** Una cadena es un palíndromo si no pierde su significado al leerla de derecha a izquierda y viceversa. Ejemplos: AMA y 12321.  
Estructure el programa en tantas funciones como sean necesarias.
7. Realizar una función que reciba dos parámetros de tipo cadena de caracteres y devuelva un valor lógico (*boolean*). La función devolverá *True* si las dos cadenas son semejantes y *False* si no lo son.  
**Nota:** Dos cadenas son semejantes si ambas se pueden leer igual al invertir el orden de los caracteres de una de ellas. Ejemplo: ROMA y AMOR.  
Estructure el programa en tantas funciones como sean necesarias.
8. Realizar un programa que simule  $m$  tiradas de un dado. Los valores de las tiradas serán números aleatorios entre 1 y 6 (ambos incluidos). La cantidad de tiradas no puede ser mayor que 20. El programa debe ser capaz de aceptar el número de tiradas, generar aleatoriamente un valor para cada tirada, mostrar los valores obtenidos en la pantalla y la frecuencia relativa de aparición de cada uno de los valores.

La frecuencia relativa del número  $i$  ( $i = 1, 2, 3, 4, 5, 6$ ) viene dada por la expresión:

$$f_r(i) = \frac{n(i)}{m}, \text{ siendo } n(i) \text{ el total de apariciones del número } i \text{ y } m \text{ es el total de lanzamientos}$$

realizados.

El programa debe cumplir los siguientes requisitos:

- a. Los números aleatorios se generan mediante un procedimiento que recibe dos parámetros: la cantidad de tiradas a realizar y una matriz donde se almacenan los valores de las tiradas.
- b. El cálculo de la frecuencia de aparición se realiza mediante una función que recibe como parámetro la matriz con los valores de las tiradas y el número cuya frecuencia se desea calcular. Esta función devuelve la frecuencia de aparición de dicho número.
- c. Al inicio del programa se pide el número total de tiradas.

Para generar números aleatorios se utiliza la función **Rnd**, cuyo prototipo se muestra a continuación y que devuelve un número aleatorio mayor o igual a 0 y menor que 1:

`Rnd[(Number)] As Single`

Si <i>Number</i> es	Rnd genera:
Menor que cero	Siempre el mismo número, utilizando <i>Number</i> como valor de inicialización.
Mayor que cero	El siguiente número aleatorio en la secuencia.
Igual a cero	El número generado por última vez.
Omitido	El siguiente número aleatorio en la secuencia.

9. Realizar una función que devuelva el resultado de la progresión geométrica siguiente:

$$1 - 2 + 3 - 4 + \dots \text{MaxN} \quad (\text{Se suman los impares y se restan los pares})$$

La función recibe como parámetro el valor MaxN.

10. Realizar un procedimiento que reciba un parámetro de tipo *Double*. Dicho procedimiento debe devolver en otros dos parámetros de tipo entero, la parte entera y la parte decimal, respectivamente.

Ejemplo: Parámetro1 = 234,5678

Después de operar, Parámetro2 = 234; Parámetro3 = 5678

11. Realizar el ejercicio 13 del boletín III de forma modular. Los módulos tendrán la siguiente estructura:

- Un procedimiento *Sub* para determinar los divisores propios de un número. Los parámetros son: el número en cuestión, una matriz en la que se guardarán los divisores propios del número y la cantidad de divisores que hay guardados en la matriz.  
Nota: Observe que los dos últimos parámetros se pasan por referencia (*ByRef*).
- Una función que devuelva la suma de los elementos de una matriz. Los parámetros son: la matriz y la cantidad de valores almacenados en la matriz.

12. Realizar una función que devuelva *True* si un número de DNI es correcto o *False* en caso contrario. La función tiene un parámetro de tipo *String*, a través del cual se le pasa el dato a comprobar.

Un número de DNI está compuesto por un número de 8 dígitos, seguidos de una letra. Se calcula el resto de la división entera entre el número y 23 y se obtiene la letra de ese resto según la siguiente tabla:

Resto	Letra	Resto	Letra	Resto	Letra	Resto	Letra
00	T	06	Y	12	N	18	H
01	R	07	F	13	J	19	L
02	W	08	P	14	Z	20	C
03	A	09	D	15	S	21	K
04	G	10	X	16	Q	22	E
05	M	11	B	17	V		

Sugerencia: Extraer los primeros 8 caracteres del DNI a una cadena temporal y aplicar la función *IsNumeric* antes de convertirla a su valor numérico.

Buscar la información de la función *IsNumeric* en el MSDN.

13. Desarrollar una función con un parámetro de tipo *String*. La función debe retornar una cadena de caracteres que sea igual que el valor pasado en el parámetro pero que intercambie los puntos por comas y viceversa.

Por ejemplo, si la función recibe como parámetro la cadena "39,889,667.345", debe retornar la cadena "39.889.667,345"

14. Desarrollar una función con los siguientes requisitos:

- Posee tres parámetros. El primero es una matriz de números reales, el segundo es la cantidad de datos que tiene almacenada la matriz y el tercero es un valor que se desea insertar dentro de la matriz.
- Los datos almacenados en la matriz están ordenados de menor a mayor.

- c. En el primer parámetro se devolverá la matriz en la cual se ha insertado el tercer parámetro.
- d. El número de elementos de la matriz debe quedar actualizado.
- e. La función retorna la posición en la cual quedó insertado el número.

Ejemplo.

Se llama a la función con los parámetros:

Matriz: 1,2;1,6;1,8;1,8;2,1;3,2;3,3;3,3;3,3;5,6;9,8;10,0;11,3

Cantidad de elementos: 13

Valor a insertar: 2,0

Al finalizar la función:

Matriz: 1,2;1,6;1,8;1,8;**2,0**;2,1;3,2;3,3;3,3;3,3;5,6;9,8;10,0;11,3

Cantidad de elementos: 14

La función devuelve el valor 4.

15.