

Algoritmos y Programación I - 75.02

Guía de Ejercicios N° 5

Funciones

Nota: Los siguientes ejercicios se refieren a programas ANSI-C modularizados.

- 1) Escribir un procedimiento que calcule el factorial de un número natural pasado como parámetro, en forma iterativa. Realizar todas las validaciones que considere necesarias.
- 2) a) Escribir una función que reciba como parámetros un arreglo de doubles y su longitud, y devuelva al máximo (o el mínimo) valor contenido en él.
b) ¿Cómo modificaría la interfaz de la función, para pasarle al módulo un token de tipo enumerativo, que indique cuál de los dos extremos se desea calcular?

Sugerencia:

```
typedef enum { MAXIMO, MINIMO } t_extremo;
```

- 3) Escribir una función que reciba un arreglo de doubles y su longitud como parámetros, y retorne el promedio aritmético de los valores contenidos en él.
- 4) Idem para el desvío estándar.
Nota: reutilice el código desarrollado en el ejercicio anterior.
- 5) a) Defina un tipo de dato "tipodato" a partir del tipo nativo "float" (usando `typedef`)
b) Defina un tipo enumerativo "tescala" que contenga los símbolos CELSIUS y FAHRENHEIT.
c) Escriba una función que reciba un valor de temperatura en precisión doble, y la escala de temperaturas de destino, y realice la conversión paramétrica del valor.
- 6) a) Defina un tipo enumerativo "tmes" con 12 símbolos que representen a cada uno de los meses del año.
b) Declare un arreglo de cadenas de caracteres con los nombres de los 12 meses del año.
c) Escriba un procedimiento que realice la traducción de tokens del tipo enumerativo "tmes" a strings, retornando la cadena de caracteres correspondiente por la interfaz al módulo invocante.
d) El diccionario de datos, ¿debería ser local a la función o una variable global?
- 7) Modularizar los ejercicios de la Guía n° 3 .
- 8) Escribir un procedimiento que reciba un arreglo de genere N elementos de tipo entero, y su longitud, y que lo retorne cargado con N números aleatorios .
- 9) Escribir un procedimiento que muestree una forma de onda determinada y almacene las N muestras en un vector de doubles recibido como parámetro, de acuerdo al siguiente prototipo:

```
void sinu_samples ( double values [], int length );
```

La forma de onda debe responder a la expresión:

$$v(t) = \text{Amp} * \text{seno} (2 * \text{PI} * \text{Frec} * t + \text{Fase})$$

en donde:

t: variable independiente (tiempo)
Frec: frecuencia de la onda senoidal, en Hz.
Fase: Fase inicial en radianes.
Amp: Amplitud pico de la onda senoidal.

- 10) Escribir una función que permita evaluar una función polinómica en un punto, a partir de un vector de coeficientes y su grado recibidos como parámetros.
El prototipo de la función pedida es:

```
double PolinomioX(double Coef[], int Grado, double Variable) ;
```

- 11) Escribir un procedimiento que, reutilizando el código desarrollado en el ejercicio anterior, genere N muestras de una función polinómica de grado G, cuyos coeficientes y grado recibe como parámetros, y retorne por la interfaz las N muestras obtenidas sobre un vector de doubles.

El prototipo de la función pedida es el siguiente:

```
void SamplePoli (
    double Muestras[],
    int    Cantidad,
    double Inicio,
    double Fin,
    double Coeficientes[],
    int    Grado) ;
```

- 12) Escribir una función que permita tabular la función de Bessel de Primera especie y orden "n".
- 13) Modificar los ejercicios 47) y 48) de la Guía N° 3 colocando los distintos diccionarios en diferentes archivos de extensión ".c", utilizando variables externas (utilizando el modificador `extern`) y compilación condicional.
- 14) ¿Es necesario incluir el nombre de los parámetros formales en el prototipo de una función? ¿Qué ventajas puede tener el hacerlo?
- 15) Indicar si el siguiente fragmento de código son correctos o no. Justicar.

```
typedef int auto; /* categoria del auto */

void f (void)
{
    auto mi_auto;

    mi_auto = 735;

}
```

- 16) Mostrar la utilización de una variable con modificador `static` dentro de una función.