

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) 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.
- 3) Idem para el desvío estándar. Nota: reutilice el código desarrollado en el ejercicio anterior.
- 4)
 - a) Defina un tipo enumerativo `mes_t` 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 `mes_t` 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?
- 5) Escribir un procedimiento que reciba un arreglo de n elementos de tipo entero, y su longitud, y que lo retorne cargado con n números aleatorios.
- 6) 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:

```
status_t sinu_samples(double values[], size_t length);
```

La forma de onda es la enunciada en el ejercicio 41) de la guía 3.

- 7) Escribir un procedimiento que, reutilizando el código desarrollado en el ejercicio 42a) de la guía 3, 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:

```
1 status_t SamplePoli (
2     double Muestras[],
3     size_t Cantidad,
4     double Inicio,
5     double Fin,
6     double Coeficientes[],
7     size_t Grado);
```

- 8) Modularizar todos los ejercicios de la Guía N°3 que aún no estén modularizados.
- 9) Escribir una función que permita tabular la función de Bessel de Primera especie y orden n .
- 10) Modificar los ejercicios 48) y 49) 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.
- 11) ¿Es necesario incluir el nombre de los parámetros formales en el prototipo de una función? ¿Qué ventajas puede tener el hacerlo?
- 12) Indicar si el siguiente fragmento de código es correcto o no. Justicar.

```
1 typedef int auto; /* categoria del auto */
2
3 void f (void)
4 {
5     auto mi_auto;
6     mi_auto = 735;
7 }
```

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