

2. Programas lineales - Tipos y Constantes

Nota: Los siguientes ejercicios introductorios se refieren a programas ANSI-C no modularizados (autocontenidos dentro de la función `main()`).

2.1. Tipos enumerativos y Constantes

- 1)
 - a) Escriba un programa que defina un tipo enumerativo con los símbolos `TRUE` y `FALSE`.
 - b) Redefina a este tipo enumerativo como tipo `bool_t`.
 - c) Declare una variable de tipo `bool_t` y asígnele el símbolo `TRUE`.
 - d) Escriba una porción de código que imprima por `stdout` una cadena de caracteres conforme el valor de la variable anterior (traducción semántica).
- 2) Idem para un tipo enumerativo `status_t` compuesto por los símbolos `ERROR` y `OK`.
- 3)
 - a) Defina un tipo enumerativo `dia_t` compuesto por 7 símbolos que representen a cada día de la semana.
 - b) Escriba una porción de código que muestre por `stdout` el contenido de una variable de tipo `dia_t` (traducción).
- 4)
 - a) Defina un tipo de dato `dato_t` a partir del tipo nativo `float` (usando `typedef`).
 - b) Defina un tipo enumerativo `escala_t` que contenga los símbolos `CELSIUS` y `FAHRENHEIT`.
 - c) Escriba un fragmento de código que lea del `stdin` un número representando un valor de temperatura en una de las dos escalas, y la convierta paramétricamente a la otra, mostrando el resultado por `stdout`.
- 5)
 - a) Defina un tipo enumerativo `mes_t` con 12 símbolos que representen a cada uno de los meses del año.
 - b) Escriba un fragmento de código que a partir del contenido de una variable de tipo `mes_t`, imprima por pantalla la descripción del mes (traducción).
- 6) Escribir un programa que lea un texto desde el teclado y calcule la cantidad de caracteres alfanuméricos, no alfanuméricos y numéricos que se ingresan. El programa debe informar esos valores por pantalla. (Hint: usar `isalnum(int character)`, `isalpha(int character)`, `isdigit(int character)`.)
- 7) Indicar si el siguiente fragmento de código es correcto o no. Justificar.

```

1  typedef enum { CORRECTO, INCORRECTO } resultado_t;
2
3  int x;
4  resultado_t proceso = CORRECTO;
5
6  if (proceso == CORRECTO) x = 1;
7  else x = -1; /* proceso incorrecto */

```

- 8) Indicar si el siguiente fragmento de código es correcto o no. Justificar.

```

1  #define CORRECTO      0
2  #define INCORRECTO    1
3
4  int resultado = CORRECTO;
5
6  if (resultado == INCORRECTO) ...

```

- 9)
 - a) Un dispositivo de comunicaciones puede utilizar una velocidad de transferencia de 1200, 2400, 4800, y 9600 baudios. Definir un tipo enumerativo que modelice dicha situación.
 - b) ¿Conviene utilizar un prefijo para los tokens del tipo enumerativo? ¿Por qué?
- 10) En relación al ejercicio anterior, dado que internamente los símbolos se representan como un número entero, ¿es posible compilar lo siguiente?

```

1  typedef enum { 1200, 2400, 4800, 9600 } baudrate_t;

```

2.2. Directivas al Preprocesador

- 11) Escribir una directiva de preprocesador para realizar cada una de las siguientes tareas:
 - a) Si la constante simbólica `TRUE` está definida, eliminarla y volverla a definir como `1`.
 - b) Idem pero sin usar la directiva `#ifdef`.
 - c) Si la constante simbólica `TRUE` está definida, eliminarla y volverla a definir como `1`, usar la directiva `#ifdef`.
- 12) Definir un token `DEBUG`, y escribir un fragmento de código que sea compilado (o no) dependiendo de si se está en modo `DEBUGing` o productivo. ¿Para qué puede servir esta construcción?
- 13) ¿Cómo se puede parametrizar el código del ejercicio 5), de forma de poder soportar varios idiomas? (usar directivas al preprocesador).

Nota: Los siguientes ejercicios introductorios se refieren a programas ANSI-C modularizados en funciones sencillas.

2.3. Tipos enumerativos y funciones rudimentarias

- 14) En el ejercicio 4, escribir una función que reciba una temperatura y una escala y devuelva la conversión a la escala restante. Utilizarla.
- 15) Un ángulo se considera agudo si es menor que 90 grados, obtuso si es mayor que 90 grados y recto si es igual a 90 grados.
 - a) Definir un tipo enumerativo para los diferentes tipos de ángulos.
 - b) Escribir una función que reciba un ángulo expresado en grados y devuelva el enumerativo correspondiente.
 - c) Escribir una función que dado un tipo enumerativo imprima el tipo de ángulo correspondiente.
 - d) Escribir un programa que acepte un ángulo en grados y muestre el tipo de ángulo correspondiente al valor. Utilizar las funciones y tipos definidos previamente.
 - e) ¿Cómo modificaría al programa anterior si el usuario ingresara el ángulo en radianes en vez de en grados?
- 16) El nivel de grado de los estudiantes de Ingeniería Electrónica que no han terminado la universidad se determina utilizando la siguiente tabla:

Número de créditos	Grado
Menos que 48	Primer año
48 a 95	Segundo año
96 a 143	Tercer año
144 o más	Cuarto año

Usando esta información, escribir un programa que acepte el número de créditos que ha acumulado un estudiante y determine en qué grado se encuentra, mostrando los resultados por pantalla. Utilizar una función para categorizar el número de créditos y otra para imprimir el año.

- 17) ¿Qué le modificaría al programa anterior si quisiera utilizarlo para la carrera de Ingeniería Mecánica, que requiere 260 créditos y no 240 (i.e. 52 créditos por año) para alcanzar el título de grado?
- 18) Cada unidad de disco de un cargamento está marcada con un código del 1 al 4, el cual indica un fabricante, como sigue:

Código de fabricante (Manufacturer ID)	Denominación (Naming)
1	3M Corporation
2	Maxell Corporation
3	Sony Corporation
4	Verbatim Corporation

Escribir un programa que acepte el número de código como dato de entrada y despliegue el nombre correcto del fabricante con base en el valor introducido (usar tipos enumerativos y constantes simbólicas para la modelización).

- 19)** ¿Qué modificaciones deberían realizarse en el programa anterior si en vez de ingresarse el código, quisieran ingresarse los símbolos **'3'**, **'M'**, **'S'** y **'V'** respectivamente?