

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

Tipos enumerativos y constantes

- 1) a) Escribir un programa que defina un tipo enumerativo con los símbolos `TRUE` y `FALSE`.
b) Redefinir este tipo enumerativo como tipo `bool_t`.
c) Declarar una variable de tipo `bool_t` y asignarle el símbolo `TRUE`.
d) Escribir una porción de código que imprima por `stdout` una cadena de caracteres según el valor de la variable anterior (traducción semántica).
- 2) Ídem para un tipo enumerativo `status_t` compuesto por los símbolos `ERROR` y `OK`.
- 3) a) Definir un tipo enumerativo `dia_t` compuesto por 7 símbolos que representen cada día de la semana.
b) Escribir una porción de código que muestre por `stdout` el contenido de una variable de tipo `dia_t` (traducción).
- 4) a) Definir un tipo de dato `dato_t` a partir del tipo nativo `float` (mediante el uso de `typedef`).
b) Definir un tipo enumerativo `escala_t` que contenga los símbolos `CELSIUS` y `FAHRENHEIT`.
c) Escribir un fragmento de código que lea del flujo de entrada estándar un número que represente un valor de temperatura en una de las dos escalas y la convierta paramétricamente a la otra, mostrando el resultado por el flujo de salida estándar.
- 5) a) Definir un tipo enumerativo `mes_t` con 12 símbolos que representen cada uno de los meses del año.
b) Escribir 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.
Sugerencia: usar las funciones de biblioteca `isalnum(int caracter)`, `isalpha(int caracter)` e `isdigit(int carácter)`.
- 7) 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. Escribir un programa que acepte un ángulo y muestre el tipo de ángulo correspondiente al valor introducido en grados. Usar tipos enumerativos para manejar los tres tipos de ángulo.
- 8) El nivel de grado de los alumnos que no han terminado los estudios en una universidad se determina mediante la siguiente tabla:

Número de créditos	Grado
< 48	Primer año
48 a 95	Segundo año
96 a 143	Tercer año
> 144	Cuarto año

Con esta información, escribir un programa que acepte el número de créditos que ha acumulado un alumno y determine en qué grado se encuentra, mostrando los resultados por pantalla.

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

Código de fabricante (<i>manufacturer id</i>)	Denominación (<i>naming</i>)
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).

- 11) ¿Qué modificaciones se deberían realizar en el programa anterior si en vez de ingresar el código, si se quisiera ingresar los símbolos “3”, “M”, “S” y “V” respectivamente?
- 12) Indicar si el siguiente fragmento de código es correcto o no. Justificar.

```
typedef enum {CORRECTO, INCORRECTO} resultado_t;

int x;
resultado_t proceso = CORRECTO;

if (proceso == CORRECTO) x = 1;
else x = -1; /* proceso incorrecto */
```

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

```
#define CORRECTO          0
#define INCORRECTO 1

int resultado = CORRECTO;

if (resultado == INCORRECTO) ...
```

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

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

Directivas al Preprocesador

16) Escribir un fragmento de código basado en una directiva al preprocesador para realizar la siguiente tarea:

Si la constante simbólica TRUE está definida, eliminarla y volverla a definir como 1.

17) Definir un token DEBUG y escribir un fragmento de código que sea compilado (o no) dependiendo de si se está en modo DEBUGging o en modo productivo. ¿Para qué puede servir esta construcción?

18) ¿Cómo se puede parametrizar el código del ejercicio 5), de forma de poder permitir varios idiomas? (usar directivas al preprocesador).