

Estructuras Decisivas o Condicionales

Si queremos ejecutar una instrucción o un grupo de instrucciones sólo cuando determinada condición es cierta usaremos la sentencia **if**:

```
if (expresión de la condición)
{bloque que se ejecutará si la condición es cierta}
```

La expresión que se coloca entre paréntesis en el **if** es una *expresión de condición*. C evalúa dicha expresión y produce un resultado que puede ser 0 u otro valor distinto (normalmente 1). Si es 0, se dice que *la condición no se cumple*. Si vale otro valor numérico distinto de 0, se dice

que *la condición se cumple*. Podemos ampliar esta sentencia añadiendo una cláusula “else” (en otro caso) para indicar qué sentencias queremos que se ejecuten si la condición no se cumple:

```
if (expresión de la condición)
{bloque que se ejecutará si la condición es cierta}
else {bloque que se ejecutará si la condición no es cierta}
```

Si sólo se ejecuta una sentencia en el primer o segundo bloque del if, se pueden omitir las llaves en ese bloque.

1. Tipee, compile y ejecute el siguiente programa:

```
main()
{ float nota;
  printf("Introduzca la nota del 0 al 10: ");
  scanf("%f", &nota);
  if (nota < 5) printf("SUSPENSO\n");
  else printf("APROBADO\n");}
```

¿qué sucede si la nota introducida es 5, menor que 5, o mayor que 5. Modifíquelo para que muestre por pantalla la palabra “SUSPENSO”, “APROBADO”, “NOTABLE” o “SOBRESALIENTE”, según sea la nota introducida.

2. Introduzca el siguiente programa que utiliza la sentencia condicional **if** y siga la traza para observar su funcionamiento. Use el depurador si no entiende que está ocurriendo, y observe los valores de las variables del programa.

```
main()
{int valor, resultado;
 printf("Introduce un numero entero:");
 scanf("%d",&valor);
 resultado = valor%2;
 if (resultado)
 printf("El numero introducido era IMPAR");
 else
 printf("El numero introducido era PAR");}
```

- a) Sustituya la expresión que da valor a la variable `resultado` por esta otra: `valor%7`. ¿Cuándo se ejecuta el primer `printf` y cuándo el segundo? Modifique el mensaje de los `printf` para que el programa sea coherente con el comportamiento de la sentencia `if`. Puede usar el depurador para observar el valor que adquiere la variable `resultado` para diferentes números introducidos por teclado en la variable `valor`.
- b) Ídem para la expresión: `valor>=0 && valor<10`
- c) Ídem para la expresión: `valor%2==0 && valor>10`
- d) Ídem para la expresión: `valor%2==0 || valor>10`
- e) Ídem para la expresión: `!(valor%2!=0 && valor<=10)`

Tips: Para el C, es verdadero cualquier valor que sea distinto de cero, siendo falso sólo cuando es igual a cero. Es por ello que tiene sentido una expresión del tipo **if (x)**, y esta será cierta siempre que *x* sea distinto de cero, independientemente del tipo *x*.

2. Tipear el siguiente programa que utiliza la construcción **else-if**. Observar su funcionamiento con el depurador si es necesario.

```
main()
{char caracter;
 printf("Introduce un carácter y pulsa Intro: ");
 scanf("%c",&caracter);
 if ((caracter>='A') && (caracter<='Z'))
 printf("La letra es una mayúscula\n");
 else if ((caracter>='a') && (caracter<='z'))
 printf("La letra es una minúscula\n");
 else
 printf("No es una letra\n");}
```

Ampliarlo para que además de decirnos si el carácter introducido es una letra mayúscula o minúscula, que nos diga si el carácter es un dígito numérico.

