Operadores aritméticos

Operador de asignación

x = 1 ;

z = 1.35 ;

Operadores aritméticos

x + + ;	equivale a	x = x+1
x- - ;	equivale a	x = x-1
x + = 8 ;	equivale a	x = x + 8
x - = 8 ;	equivale a	x = x – 8

Operadores relacionales (izquierda) y lógicos (derecha) en C

Mayor que	<	Conjunción ò Y lógicos	&&
Menor que	>	Disyunción u O lógicos	||
Menor o igual que	<=	Negacion ò NO lógico	!
Igual que	==		
Distinto que	!=		

Operadores matemáticos unarios (++ , --) y binarios

Signo negativo	-
Incremento	++
Decremento	--
Multiplicacion	*
Division	/
Modulo	%
Suma	+
Resta	-

Tabla: Las secuencias de escape más frecuentemente usadas

Secuencia	Significado
\a	Campana (alerta)
\b	Retroceso
\n	Nueva línea
\t	Tabulador horizontal
\\	Diagonal inversa
\?	Signo de interrogación
\'	Comilla simple
\"	Comilla doble

Ejemplo: Operadores unarios

```
1:    /* Demuestra los modos de prefijo y posfijo de operadores unarios */
2:
3:    #include <stdio.h>
4:
5:    int a, b;
6:
7:    main()
8:    {
```

```
9:        /* Pone a y b igual a 5 */
10:
11:       a = b = 5;
12:
13:       /* Los imprime decrementándolos cada vez */
14:        /* Usa modo de prefijo para b y modo de posfijo para a */
15:
16:       printf("\n%d   %d", a--, --b);
17:       printf("\n%d   %d", a--, --b);
18:       printf("\n%d   %d", a--, --b);
19:       printf("\n%d   %d", a--, --b);
20:       printf("\n%d   %d", a--, --b);
21:
22:       return 0;
23:  }
```

Ejemplo: Precedencia de los operadores lógicos

```
1:  #include <stdio.h>
2:
3:  /* Inicializa variables. Observe que c no es menor que d, */
4:  /* que es una de las condiciones que se han de probar. */
5:  /* Por lo tanto, la expresión completa debe evaluar a falso */
6:
7:  int a = 5, b = 6, c = 5, d = 1;
8:  int x;
9:
10: main()
11: {
12:     /* Evalúa la expresión sin paréntesis */
13:
14:     x = a < b || a < c && c < d;
15:     printf"\nWithout parentheses the expression evaluates as %d", x),
16:
17:     /* Evalúa la expresión con paréntesis */
18:
19:     x = (a < b || a < c) && c < d;
20:     printf("\nWith parentheses the expression evaluates as \
              %d", x);
21:     return 0;
22: }
```