**Operadores Bitwise ( a nivel del Bit ) en C**

conjunción Lógica o multiplicación Booleana a nivel del bit **AND &**

a = 10010101

b = 00110100

a & b = 00010100

disyunción Lógica o suma Booleana a nivel del bit **OR |**

a = 10010101

b = 00110100

a | b = 10110101

Disyunción exclusiva Lógica a nivel del bit **XOR ^**

a = 10010101

b = 00110100

a ^ b = 10100001

negación Lógica a nivel del bit **NOT ~**

a = 10010101

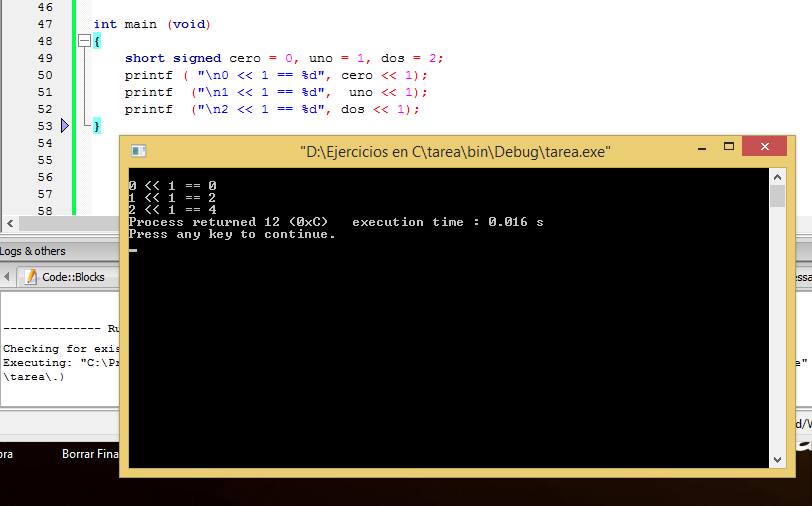
~a = 01101010

Operación **<<**

Sea a una cadena de bits, n un entero, a << n, devuelve la misma cádena movida n bits a la izquierda ( perdiendose los primeros n bits de la cadena ) y reemplazando las n posiciones a la derecha por 0.

a = 10010101

a << 2 = 01010100

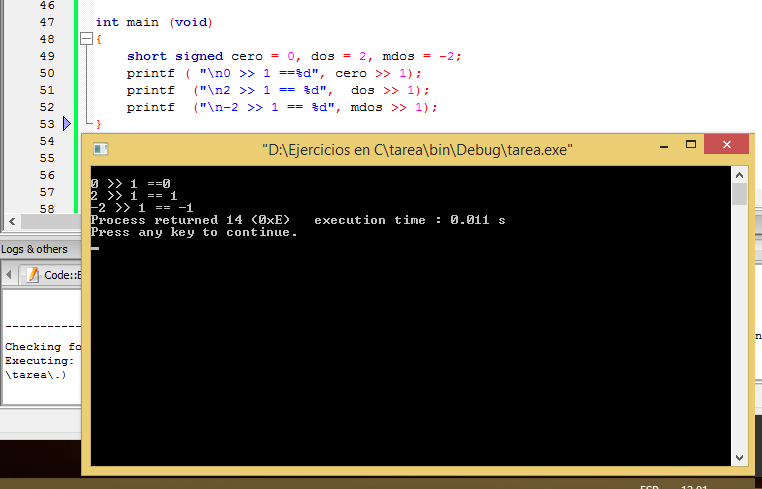


Operación **>>**

Sea a una cadena de bits, n un entero, a >> n, devuelve la misma cadena movida n bits a la derecha ( perdiéndose los últimos n bits de la cadena ) y reemplazando las n posiciones a la izquierda por 0.

a = 10010101

a >> 3 = 00010010



Recordar que:

Si los operandos no son enteros el compilador realiza la conversión pertinente

No confundir los **operadores de bits**,  **&** y **|**, con los **operadores lógicos** **&&** y **||**

**&** puede ser también el operador de referencia de punteros

Asignaciones compuestas con operadores de bits

x &= y; // equivale a: x = (x & y);

x ^= y; // equivale a: x = (x ^ y);

x |= y; // equivale a: x = (x | y);

x <<= y; // equivale a: x = (x << y);

x >>= y; // equivale a: x = (x >> y);