

OPERADORES LOGICOS Y BINARIOS

OPERADORES LOGICOS, el resultado es True o False

AND (&&)

OPERACION	RESULTADO
$((2 > 3) \&\&(1 < 5))$	False

OR (||)

OPERACION	RESULTADO
$((2 > 3) (1 < 5))$	True

Identidad (==)

OPERACION	RESULTADO
$(2 == 3)$	False

Diferente (!=)

OPERACION	RESULTADO
$(2 != 3)$	True

Exclusive OR XOR (^)

OPERACION	RESULTADO
$((5 > 3) ^ (1 < 5))$	False

OPERADORES BINARIOS, realiza operaciones entre números a nivel binario.

Debemos saber que si b = Número de bits; un número escrito con b bit's debe estar en el intervalo $[0, 2^b - 1]$

Es decir, si un número de ser escrito con $b=8$ bit's; entonces deberá estar en $[0, 2^8 - 1]$; lo que significa $[0, 255]$.

Sea el byte:

Conversión de Decimal a Binario:

Potencia 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valor Decimal	128	64	32	16	8	4	2	1

Bit	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
-----	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Obsérvese que cada posición en el byte (8 bit's) tiene un valor decimal asociado; y además por ser de 8 bit's; solo se podrá escribir números del intervalo [0,255].

Ejemplo:

Convertir el número 187 a binario:

El número 187 es menor a 255 entonces puede ser escrito en 8 bit's.

Paso 1, veamos el byte de 8 bit's:

Valor Decimal	128	64	32	16	8	4	2	1
Bit	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀

El Bit b₇ tiene un valor decimal asociado de 128 que es inferior a 187; por lo tanto debe estar el 1 en b₇.

Valor Decimal	128	64	32	16	8	4	2	1
Bit	1	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀

Pero quedaría: 187-128=59

Paso 2, el bit que sigue sería b₆ pero tiene un valor decimal asociado de 64 que es superior a 59; por lo tanto debe estar el 0 en b₆

Valor Decimal	128	64	32	16	8	4	2	1
Bit	1	0	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀

Paso 3, el bit que sigue sería b₅ pero tiene un valor decimal asociado de 32 que es inferior a 59; por lo tanto debe estar el 1 en b₅

Valor Decimal	128	64	32	16	8	4	2	1
Bit	1	0	1	b ₄	b ₃	b ₂	b ₁	b ₀

Pero quedaría: $59-32=27$

Paso 4, el bit que sigue sería b_4 pero tiene un valor decimal asociado de 16 que es inferior a 27; por lo tanto debe estar el 1 en b_4

Valor Decimal	128	64	32	16	8	4	2	1
Bit	1	0	1	1	b_3	b_2	b_1	b_0

Pero quedaría: $27-16=11$

Paso 5, el bit que sigue sería b_3 pero tiene un valor decimal asociado de 8 que es inferior a 11; por lo tanto debe estar el 1 en b_3

Valor Decimal	128	64	32	16	8	4	2	1
Bit	1	0	1	1	1	b_2	b_1	b_0

Pero quedaría: $11-8=3$

Paso 6, el bit que sigue sería b_2 pero tiene un valor decimal asociado de 4 que es superior a 3; por lo tanto debe estar el 0 en b_2

Valor Decimal	128	64	32	16	8	4	2	1
Bit	1	0	1	1	1	0	b_1	b_0

Paso 7, el bit que sigue sería b_1 pero tiene un valor decimal asociado de 2 que es inferior a 3; por lo tanto debe estar el 1 en b_1

Valor Decimal	128	64	32	16	8	4	2	1
Bit	1	0	1	1	1	0	1	b_0

Pero quedaría: $3-2=1$

Paso 8, el bit que sigue sería b_0 pero tiene un valor decimal asociado de 1 que es igual al 1 que nos queda; por lo tanto debe estar el 1 en b_0

Valor Decimal	128	64	32	16	8	4	2	1
Bit	1	0	1	1	1	0	1	1

Convertir un número binario a base 8:

(Sistema Octal) (0, 1, 2, 3, 4, 5, 6, 7)

Dado el número binario 10111011, encontrar su representación Octal.

Primero se escribe el número:

1	0	1	1	1	0	1	1
---	---	---	---	---	---	---	---

Luego se agrupa de a 3 bit's a partir del bit menos significativo (b_0); si faltasen bit se completan con 0

0	1	0	1	1	1	0	1	1
---	---	---	---	---	---	---	---	---

Luego se trata cada terna de números como si fuesen binarios independientes y se convierten a decimal:

b2

4	2	1
0	1	0

$0*1+1*2+0*4$
2

b1

4	2	1
1	1	1

$1*1+1*2+1*4$
7

b0

4	2	1
0	1	1

$1*1+1*2+0*4$
3

El número en Octal será: 273_8 en C++ se escribe 0273 (Debe tener un 0 al inicio)

Convertir un número binario a base 16:

(Sistema Hexadecimal) (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

Dado el número binario 10111011, encontrar su representación Hexadecimal.

Primero se escribe el número:

1	0	1	1	1	0	1	1
---	---	---	---	---	---	---	---

Luego se agrupa de a 4 bit's a partir del bit menos significativo (b_0); si faltasen bit se completan con 0

1	0	1	1	1	0	1	1
---	---	---	---	---	---	---	---

Luego se trata cada terna de números como si fuesen binarios independientes y se convierten a decimal:

b1			
8	4	2	1
1	0	1	1
$1*1+1*2+0*4+1*8$			
B			

b0			
8	4	2	1
1	0	1	1
$1*1+1*2+0*4+1*8$			
B			

El número en Hexadecimal será: BB_{16} en C++ se escribe 0xBB (Debe tener un 0x al inicio)

Operadores Binarios

AND (&)

OPERACION	RESULTADO
0 & 0	0
0 & 1	0
1 & 0	0
1 & 1	1
(1011)& (1100)	1000

OR (|)

OPERACION	RESULTADO
0 0	0
0 1	1
1 0	1
1 1	1
(1011) (1100)	1111

Desplazamiento a la Derecha (>>)

OPERACION	RESULTADO
(10111011)>> 3	00010111011
	Resultado: 00010111

Desplazamiento a la Izquierda (<<)

OPERACION	RESULTADO
(10111011)>> 5	1011101100000
	Resultado: 01100000