



Escuela de
Ciencia y Tecnología
ECyT_UNSAM

Programación (en C)

Primer Cuatrimestre 2025

programacionbunsam@gmail.com



Sistemas de numeración y Bitwise operations



Numeración Binaria

- Sistema de Base-2 → solo usa 2 dígitos, **0** y **1**
- Valores posicionales (el valor del dígito depende de su posición).
- Cada dígito individual se llama bit.

Numeración Hexadecimal

- Base 16 → 16 dígitos: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.
- Cada dígito representa un número del 0 al 15
- Cada dígito hexa puede ser representado por 4 bits (i.e. F = 1111).



Numeración BCD (Binary Coded Decimal)

- Representación Decimal → uso 4 bits para representar un número del 0 al 9 (siempre me sobran 6 combinaciones).
- Cada dígito decimal es tratado independientemente.
- Por Ejemplo: 349 = 0011 0100 1001.

Numeración Decimal

- Sistema de base 10 → 0,1,2,3,4,5,6,7,8,9
- Es el sistema de numeración más usado



Conversiones entre sistemas

MSB (most significant bit)

LSB (least significant bit)

Bin a Dec → $1\ 1\ 0\ 1 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$

Hex a Bin → $A3F = 1010\ 0011\ 1111$

Hex a Dec → $A3F = 10 \times 16^2 + 3 \times 16^1 + 15 \times 16^0 = 2623$

Dec a BCD → $157 = 0001\ 0101\ 0111$

obs. Las conversiones de BCD a Bin y a Hex se realizan pasando por el sistema decimal.



Bitwise Operations (bit a bit)

Operador **AND** (&) $\rightarrow Y = A \& B$



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Operador **OR** (|) $\rightarrow Y = A | B$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Operador **NOT** (~) $\rightarrow B = \sim A$



A	B
0	1
1	0

Operador **XOR** (^) $\rightarrow Y = A \wedge B$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



Ejemplo de código

```
int a = 5; // 0101 en binario
int b = 3; // 0011 en binario

int and_result = a & b; // Resultado: 1 (0001 en binario)
int or_result = a | b; // Resultado: 7 (0111 en binario)
int not_result = ~a; // Resultado: -6 (complemento A1 de 5 - 1010)
```

https://en.wikipedia.org/wiki/Ones%27_complement → complemento A1



Máscaras y Shifts (1)

Las máscaras son números binarios que se usan para aislar, setear, limpiar o togglear bits específicos en otro número binario.

En general son utilizadas en bitwise operations para manipular bits específicos de un integer.

- Aislando bits: para extraer bits específicos de un número, comúnmente utilizando el operador `&`.
- Seteando y Limpiando bits: pongo en 0 o en 1 un bit específico de otro número (por ejemplo, un registro en un microcontrolador).
 - ◆ Pueden ser realizadas con operadores `|` tanto como `&`.
- Toggle bits: invierto el valor de un bit específico en un número. Puede ser realizado con el operador `^`.



Máscaras y Shifts (2)

Los shifts son un tipo de operación en la que uno *mueve* los bits de un número. Hay 2 tipos de shift-operations:

❖ Left Shift → '<<'

➤ (0000 0001 << 4) == 0001 0000.

➤ (0000 0110 << 2) == 0001 1000.

❖ Right Shift → '>>'

➤ (0100 1111 >> 2) == 0001 0011.

obs. En el caso de que al shiftear bits el tamaño del número exceda el permitido, los bits que se salgan serán descartados.



Ejemplo (1)

```
#include <stdio.h>

int main() {
    int x = 5; // 0101 en binario
    int mask = 1; // 0001 en binario

    if (x & mask) {
        printf("El bit menos significativo es 1\\n");
    } else {
        printf("El bit menos significativo es 0\\n");
    }

    return 0;
}
```

En C, cualquier número distinto de 0, es considerado True (verdadero)



Ejemplo (2)

```
/* Apagar un bit en particular */
```

```
001000010
```

```
&~ 001000000
```

```
-----
```

```
000000010
```

```
/* Encender un bit en particular */
```

```
001000010
```

```
| 000100000
```

```
-----
```

```
001100010
```

&~ 001000000 =

& (~001000000) =

& 110111111



Bonus Track (%)

El `%` realiza la operación de “*módulo*”. Es utilizada para conocer el resto de la división de números enteros.

Dije que él `%` se puede usar para saber si un número es par. ¿Si quisiera saber si un número es par usando máscaras y operaciones binarias, como haría?

```
#include <stdio.h>

int main() {
    int num = 35;
    int den = 15;

    int resto = num % den; //devuelve 5

    return 0;
}
```