

CIENCIA DE LA COMPUTACION

LABORATORIO C

Leonardo Daniel Valdivia Ramos

1 BITWISE OPERATORS

Los operadores de bits son 6: $\&$, $|$, \gg , \ll , \wedge y \sim .

1.1 Operador $\&$ (AND)

Con dos números hace AND en cada bit de dos números. El resultado de AND es 1 solo si ambos bits son 1.

Ejm:

```
#include <iostream >
using namespace std;
int main()
{
    int a,b;
    a=5;//0101
    b=12;//1100
    cout<< (a&b) <<endl;
}
```

OUTPUT:

4 (0100 en binario)

1.2 Operador $|$ (OR)

Toma dos números y lo hace OR en cada bit de dos números. El resultado de OR es 1 siempre que cualquiera de los dos bits sea 1.

Ejm:

```
#include <iostream >
using namespace std;
int main()
```

```

{
    int a,b;
    a=5;//0101
    b=12;//1100
    cout<< (a|b) <<endl;
}

```

OPUTPUT:
13 (1101 en binario)

1.3 Operador ^ (XOR)

También llamado OR exclusivo toma dos números y hace XOR en cada bit de dos números. El resultado de XOR es 1 siempre que los dos bits que se estén operando sean diferentes.

Ejm:

```

#include <iostream >
using namespace std;
int main()
{
    int a,b;
    a=5;//0101
    b=12;//1100
    cout<< (a^b) <<endl;
}

```

OPUTPUT:
9 (1001 en binario)

1.4 Operador <<(desplazamiento a la izquierda)

Con un numero opera los bits moviéndolos hacia la izquierda y remplazando los lugares vacíos por 0.

Ejm:

```

#include <iostream >
using namespace std;
int main()
{
    int a,b;
    a=5;//0101
    b=12;//1100
    cout<< (a<<1) <<endl;
}

```

OUTPUT:

10 (1000 en binario)

1.5 Operador >>(desplazamiento a la derecha)

Opera los bits de un numero desplazándolos a la derecha la cantidad de deseada y remplazándolos por 0.

Ejm:

```
#include <iostream >
using namespace std;
int main()
{
    int a,b;
    a=5;//0101
    b=12;//1100
    cout<< (b>>2) <<endl;
}
```

OUTPUT:

3 (0011 en binario)

1.6 Operador ~ (negación a nivel de bits)

En un entero con signo bitwise da como resultado el inverso restado en 1, es decir $x=-x-1$

```
#include <iostream >
using namespace std;
int main()
{
    int a,b;
    a=5;//0101
    b=12;//1100
    cout<< (~a) <<endl;
}
```

OUTPUT:

-6 (1010 que es el complemento a 2 del numero 6)