

Tarea 1: Corrimiento de bits en c.

Desplazamiento de Bits

Este tema no es facil de entender al principio, pero para que quede claro, uno puede hacer corrimientos de N Bits, con $N \in \text{Naturales}$, es decir $N = 1, 2, \dots$. Esta operación sirve básicamente para modificar un número a nivel Bit. Los corrimientos de Bits, pueden ser en dos direcciones:

Corrimiento hacia la derecha

Corrimiento hacia la izquierda

Desplazamiento de Bits a la Derecha (>>)

Supongamos que tenemos un Byte llamado $a = 01101001 = 105|_{10}$, a este número se le puede ser corrimientos de bits de la siguiente manera

Número	Cantidad de Bits de Corrimiento	Resultado
01101001	1	00110100 $\Rightarrow 52 _{10}$
01101001	2	00011010 $\Rightarrow 26 _{10}$
01101001	3	00001101 $\Rightarrow 13 _{10}$
01101001	4	00000110 $\Rightarrow 6 _{10}$
01101001	5	00000011 $\Rightarrow 3 _{10}$

Por lo tanto la función para aplicarlo en el Lenguaje C, es:

numero >> cantidad bits de corrimiento;

Ejemplo: Desplazamiento a la Derecha de Bits

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char a = 105;
6     char resultado;
7
8     printf(' 'a = %d',a);
9
10    resultado = a >> 1;
11
12    printf(' 'Resultado = %d',resultado) /*imprime 52*/;
13
14    resultado = a >> 2;
15
16    printf(' 'Resultado = %d',resultado) /*imprime 26*/;
17    return 0;
18 }
```

En pocas palabras, los corrimientos de Bits hacia la derecha, consisten en realizar la división Entera del número al cual se le realiza dicho corrimiento, es decir:

numero >> cantidad bits de corrimiento; \approx numero % (cantidad bits de corrimiento + 1);

Desplazamiento de Bits a la Izquierda (< <)

Supongamos que tenemos un Byte llamado a = 01101001 \Rightarrow 105| 10, a este número se le puede ser corrimientos de bits de la siguiente manera:

01101001	1	11010010 \Rightarrow 210 ₁₀
01101001	2	10100100 \Rightarrow 420 ₁₀
01101001	3	01001000 \Rightarrow 840 ₁₀
01101001	4	10010000 \Rightarrow 1680 ₁₀
01101001	5	00100000 \Rightarrow 3360 ₁₀

Nota: Pueden apreciar que los valores obtenidos en la base decimal no coinciden en lo absoluto con su representación binaria, esto se debe al overflow que existe al realizar un desplazamiento de bits, lo cual para realizar esto, hay que tener sumo cuidado!! Por lo tanto la función para aplicarlo en el Lenguaje C , es:

numero << cantidad bits de corrimiento

```
1
2 #include <stdio.h>
3
4 int main()
5 {
6     char a = 105;
7     char resultado;
8
9     printf( ' 'a = %d' ',a );
10
11     resultado = a << 1;
12
13     printf( ' 'Resultado = %d' ',resultado ) /*imprime 210*/;
14
15     resultado = a << 2;
16
17     printf( ' 'Resultado = %d' ',resultado ) /*imprime 420*/;
18     return 0;
19 }
```