

Módulo	0785 - Programação em C/C++ - Formas Complexas
Local	
Sessão	
Formador	
Ficha	1 - Outros Sistemas Numéricos

Regras de Conversão

As regras seguintes demonstram como fazer a conversão entre os vários sistemas apresentados anteriormente.

Binário e Decimal

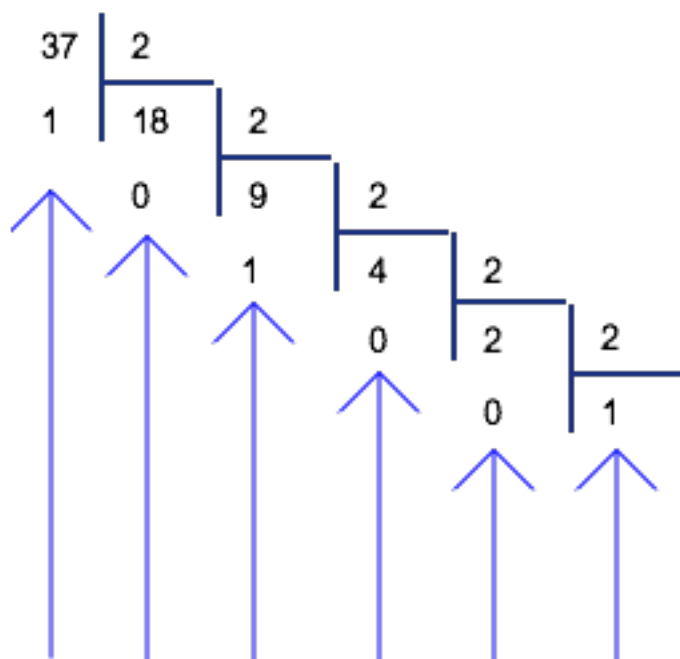
Converter de binário para decimal:

10010100101

$$(1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) =$$

$$(1 \times 32) + (0 \times 16) + (0 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1) = 37$$

Converter de decimal para binário:



100101, obtido recolhendo os restos, da direita para a esquerda

Octal e Decimal

Converter octal para decimal:

112

$$(1 \times 8^2) + (1 \times 8^1) + (2 \times 8^0) = 74$$

Converter decimal para octal:

$$\begin{array}{rcl}
 900 \% 8^3 = 1 & & 900 - ((8^3) * 1) = 338 \\
 338 \% 8^2 = 6 & \xleftarrow{\text{green}} & 338 - ((8^2) * 6) = 4 \\
 4 \% 8^1 = 0 & \xleftarrow{\text{red}} & 4 - ((8^1) * 0) = 4 \\
 4 \% 8^0 = 4 & \xleftarrow{\text{blue}} &
 \end{array}$$

Hexadecimal e Decimal

Converter decimal para hexadecimal:

A conversão entre decimal e hexadecimal é mais fácil se for feita indirectamente, isto é, se convertermos primeiro para binário e depois para hexadecimal.

Converter hexadecimal para decimal:

C0E716
 $C0E716 = (12 \times 16^5) + (0 \times 16^4) + (14 \times 16^3) + (7 \times 16^2) + (1 \times 16^1) + (6 \times 16^0) =$
 $(12 \times 4096) + (0 \times 256) + (14 \times 16) + (7 \times 1) = 49383$

Binário e Hexadecimal

Converter de binário para hexadecimal:

Separe o número binário em conjuntos de quatro dígitos, adicione zeros à esquerda se for necessário, depois basta substituir os conjuntos binários pela sua representação.

1100101
 $1100101 = 0110\ 0101 = 6\ 5 = 65$

Converter de hexadecimal para binário:

Basta fazer a operação inversa à anterior, separando os dígitos e substituindo pela sua representação binária.

E7
 $E7 = E\ 7 = 1110\ 0111 = 11100111$

Binário e Octal

A conversão de binário para octal segue a mesma regra da conversão para hexadecimal com a diferença de que os bits são agrupados três a três em vez de quatro a quatro.

Converter binário para octal:

1100101
 $1100101 = 001\ 100\ 101 = 1\ 4\ 5 = 145$

Converter octal para binário:

65
65 = 110 101

1. Execução de Exemplo

Copie o seguinte código e execute o mesmo para verificar o seu resultado. O resultado é o esperado? Porquê?

```
#include <stdio.h>

struct bf
{
    int x:4;
    int y:4;
    int z: 2;
};

int main()
{
    struct bf bits;

    bits.x = 100;
    bits.y = 2;
    bits.z = 1;

    printf("Valor de X: %d\nValor de Y: %d\nValor de Z: %d", bits.x, bits.y, bits.z);

    return 0;
}
```

- a. Escreva uma função que guarde um carácter com apenas 8 bits.

2. Converta os Seguintes Números

Binário	Decimal	Octal	Hexadécimal
001			
011			
111			
00001			
11000011			
10010			
111011			
1100100			
11001111			
101			
100			
1101			
10001			
11110			

Binário	Decimal	Octal	Hexadécimal
1010100			
1010			
110011			
101111			
10001			
1111			
10000111			
	5		
	59		
	519		
	522		
	531		
	532		
	546		
	567		
	5100		
	5255		
	5234		
	529		
	559		
	534		
	565		
	5334		
	566		
	5112		
	578		
	589		
	590		
	550		
			DEAD
			AF
			05
			12
			57
			0C

Binário	Decimal	Octal	Hexadécimal
			9FF2
			A9
			5E
			123456789ABCDEF
			F0E1D2
			C3B4
			A596
			B0C
			1000
			AA
			123F
			FEDCBA
			D0
			BED0
			B8C
			FFFFFFFA
			10
			BABE
			BAD
			732
			801
			ECA8
			379DF

3. Aritmética Binária

Resolva as seguintes operações de soma:

a. $101 + 10 =$

b. $111 + 10 =$

c. $1010 + 110 =$

d. $1010 + 1010 =$

e. $1010 + 1111 =$

f. $1011 + 101 =$

g. $11110 + 1110 =$

h. $100110 + 1101 =$

i. $101110 + 101 =$

j. $101 + 11 =$

k. $111 + 111 =$

l. $1010 + 1010 =$

m. $11101 + 1010 =$

n. $11111 + 11111 =$