

Tabela Resumo de Regras de Conversão entre algumas das Bases Numéricas Fundamentais na Computação

Conversão da base x → y \ Ação para números do conjunto:	Conjunto dos Naturais (N)	Conjunto dos Inteiros (Z) - Três possíveis representações: sinal-magnitude, complemento de 1 e complemento de 2. Existem outras... Observação: É absolutamente necessário nas representações citadas definir exatamente o tamanho da representação, em bits.	Conjunto dos racionais positivos (Q+)
10 → n	Realizar divisões sucessivas por n até quociente = 0, resultado são os restos das divisões tomados na ordem inversa. Ex: (22) <sub>10</sub> = (x) <sub>n</sub> , 22/3=7 resto 1, 7/3=2 resto 1, 2/3=0 resto 2. Logo, x=(211) <sub>3</sub> .	10 → Complemento de 2, ou 2's	Separar partes inteira e fracionária. Parte inteira (antes da vírgula) converte como feito para número em 10 → n. Parte fracionária (depois da vírgula) converte usando multiplicações sucessivas da parte fracionária pela base de destino. Após cada multiplicação, o que houver à esquerda da vírgula é convertido para um dígito na base destino. Este é acrescentado como o próximo dígito à direita, na parte fracionária do resultado. Para a próxima multiplicação, toma-se apenas a parte fracionária da multiplicação que acabou de ser realizada. Ex: (26,36) <sub>10</sub> = (x) <sub>n</sub> , 26/2=13 resto 0, 13/2=6 resto 1, 6/2=3 resto 0, 3/2=1 resto 1, 1/2=0 resto 1. Logo, a parte inteira do resultado é 11010. A parte fracionária é obtida multiplicando 0,36 sucessivamente por 2 e tomando a parte inteira de cada resultado, ou seja: 0,36*2=0,72 (0), 0,72*2=1,44 (1), 0,44*2=0,88 (0), 0,88*2=1,76 (1), 0,76*2=1,52 (1)... Parei (arbitrariamente na 5a casa depois da vírgula. Logo, tem-se que x=(11010,01011) <sub>2</sub> .
n → 10	Efetuar a soma dos produtos dos dígitos do número na base n pela base n elevada ao expoente correspondente, iniciando pelo dígito menos significativo d <sub>n-1</sub> *n <sup>n-1</sup> , d <sub>n-2</sub> *n <sup>n-2</sup> , ..., d <sub>0</sub> *n <sup>0</sup> . Ex: (734) <sub>8</sub> =(x) <sub>10</sub> . 4*8 <sup>3</sup> +3*8 <sup>2</sup> +7*8 <sup>1</sup> +4*8 <sup>0</sup> =(4+24+48) <sub>10</sub> . Logo, x=(476) <sub>10</sub> .	Complemento de 2 ou 2's → 10	Efetuar a soma dos produtos dos dígitos do número na base n pela base n elevada ao expoente correspondente, iniciando pelo dígito mais significativo com peso negativo -d <sub>n-1</sub> *2 <sup>n-1</sup> , d <sub>n-2</sub> *2 <sup>n-2</sup> , ..., d <sub>0</sub> *2 <sup>0</sup> . Ex: (1100011) <sub>2</sub> . Ex: (1100011) <sub>2</sub> = -1*2 <sup>6</sup> +1*2 <sup>5</sup> +0*2 <sup>4</sup> +0*2 <sup>3</sup> +0*2 <sup>2</sup> +1*2 <sup>1</sup> +1*2 <sup>0</sup> = -64+32+0+0+0+2+1) <sub>10</sub> . Logo, x=(-29) <sub>10</sub> .
2 → 16	A partir da direita, converter cada 4 bits para o dígito hexadecimal equivalente: 0000 → 0, 0001 → 1, ..., 1001 → 9, 1010 → A, ..., 1111 → F. Ex: (110011101) <sub>2</sub> =(x) <sub>16</sub> . 1 1001 1101. Logo, x=(19D) <sub>16</sub> .	10 → Complemento de 1 ou 1's	Separar as partes antes e depois da vírgula. Para a parte à esquerda da vírgula, a partir da direita, converter cada 4 bits para o dígito hexadecimal equivalente: 0000 → 0, 0001 → 1, ..., 1001 → 9, 1010 → A, ..., 1111 → F. Para a parte à direita da vírgula faça o mesmo a partir da esquerda. Ex: (110011,101) <sub>2</sub> =(x) <sub>16</sub> . 11 0011 , 1010). Logo, x=(33,A) <sub>16</sub> .
2 → 8	A partir da direita, converter cada 3 bits para o dígito octal equivalente: 000 → 0, 001 → 1, 010 → 2, ..., 110 → 6, 111 → 7. Ex: (10011101) <sub>2</sub> =(x) <sub>8</sub> . 10 011 101. Logo, x=(235) <sub>8</sub> .	Complemento de 1 ou 1's → 10	Separar as partes antes e depois da vírgula. Para a parte à esquerda da vírgula, a partir da direita, converter cada 3 bits para o dígito octal equivalente: 000 → 0, 001 → 1, 010 → 2, ..., 110 → 6, 111 → 7. Para a parte à direita da vírgula faça o mesmo a partir da esquerda. Ex: (1101,1101) <sub>2</sub> =(x) <sub>8</sub> . 1 101 , 110 100). Logo, x=(15,64) <sub>8</sub> .
8 → 2	Converter cada dígito octal em exatamente três bits: 0 → 000, 1 → 001, 2 → 010, ..., 6 → 110 e 7 → 111. Ex: (247) <sub>8</sub> =(x) <sub>2</sub> . 10 100 111. Logo, x=(10100111) <sub>2</sub> .	10 → Sinal Magnitude ou SM	Converter cada dígito octal em exatamente três bits: 0 → 000, 1 → 001, 2 → 010, ..., 6 → 110 e 7 → 111. Ex: (247,54) <sub>8</sub> =(x) <sub>2</sub> . 010 100 111, 101 100. Logo, x=(10100111,01101) <sub>2</sub> .
8 → 16	Converter o octal para binário como na conversão 8 → 2 e depois reagrupar bits de 4 em 4 a partir da direita, como na conversão 2 → 16. Ex: (1473) <sub>8</sub> =(x) <sub>16</sub> . 1 100 111 011 = 11 0011 0111. Logo, x=(338) <sub>16</sub> .	Sinal Magnitude ou SM → 10	Converter o octal para binário como na conversão 8 → 2 (Q+) e depois reagrupar bits de 4 em 4 a partir da direita, como na conversão 2 → 16 (Q+). Ex: (214,72) <sub>8</sub> =(x) <sub>16</sub> . 10 001 100, 111 011 010 = 10001100,11101000). Logo, x=(8C,E8) <sub>16</sub> .
16 → 2	Converter cada dígito hexadecimal em exatamente quatro bits: 0 → 0000, 1 → 0001, ..., 8 → 1000, 9 → 1001, A → 1010, ..., E → 1110 e F → 1111. Ex: (5C72) <sub>16</sub> =(x) <sub>2</sub> . 101 1100 0111 0010. Logo, x=(101110001110010) <sub>2</sub> .	16 → 2	Converter cada dígito hexadecimal em exatamente quatro bits: 0 → 0000, 1 → 0001, ..., 8 → 1000, 9 → 1001, A → 1010, ..., E → 1110 e F → 1111. Ex: (5C,72) <sub>16</sub> =(x) <sub>2</sub> . 0101 1100, 0111 0010. Logo, x=(10111000,01110010) <sub>2</sub> .
16 → 8	Converter da base hexadecimal para base 2 como em 16 → 2. Depois, reagrupar os bits gerados de 3 em 3 a partir da direita, convertendo cada um destes como na conversão 2 → 8. Ex: (617) <sub>16</sub> =(x) <sub>8</sub> . 1000 0001 0111 = 100 000 010 111. Logo, x=(4027) <sub>8</sub> .	16 → 8	Converter da base hexadecimal para base 2 como em 16 → 2 (Q+). Depois, reagrupar os bits gerados de 3 em 3 a partir da direita, convertendo cada um destes como na conversão 2 → 8. Ex: (61,7C) <sub>16</sub> =(x) <sub>8</sub> . 010 0001, 0111 1100 = 1 100 001, 011 111. Logo, x=(141,37) <sub>8</sub> .