Conversão da base x -		Tabela	Resumo de Regras de Conversão entre algumas das Bases Numéricas Fundamenta Conjunto dos Inteiros (Z) - Três possíveis representações: sinal-magnitude,	is na Computação	
> y \ Ação para			complemento de 1 e complemento de 2. Existem outras Observação: É		
números do conjunto:	Conjunto dos Naturais (N)		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>31</sub> × (x) <sub>2</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	Se negativo, guardar o sinal, assumir que o número é positivo e converter como para N, mantendo o tamanho da representação. Se positivo, o resultado estará pronto. Se negativo, aplicar a operação de troca de sinal (iniciando a direita, copiair todos os bites os 0. Depois codar o primeiro 1. Depois deste 1, inverter todos os bits seguintes até o fim da representação). Ex: (57) <sub>1/2</sub> *(1/2, x=n tiso: 75/2-28 teos). 2/2-14 resto 0, 1/2-27 resto 0, 1/2-2 resto 1, 3/2-1 resto 1, 1/2-07 resto	10-→2	Separar partes inteira e fracionária. Parte inteira (antes da virgula) converte como felto para número em 10 -> n. Parte fracionária (depois da virgula): converte usando multiplicações sucessivas da parte fracionária pela base de destino. Após cada multiplicação, que houver à exquerda da virgula é convertedo para um digito na base de estino. Este de acrescentado como o próximo digito à direita, na parte fracionária da multiplicação, que aproxima multiplicação, tomas a penas a parte fracionária da multiplicação, tomas a penas a parte fracionária da multiplicação que ascuba de ser realizada. Ezt (28,58) <sub>10</sub> (s.), 26/2-13 resto 0, 3/2-1 resto 1, 1/2-0 resto 1. Logo, a parte inteira do resultado da parte inteira de cada resultado, ou seja: 0,36°2-0,72 (0), 0,272*2-1,44 (1), 0,472*-0,88 (0), 887*2-1,5 (1), Parei (arbitrariamente na Sa casa depois da virgula. Logo, tem-se que x=(11010,01011) <sub>2</sub> .
n → 10	Efetuar a soma dos produtos dos digitos do número na base n pela base n elevada ao expoente correspondente, iniciando pelo digito menos significativo d.; 47°, 47°, 47°, 47°, 47°, 47°, 47°, 47°,	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 $-d_{a_1}^{-1} x^{2-1} + d_{a_2}^{-1} n^{-1} + + d_3^{-1} n^{-1} + d_3^{-1} n^{-1} + d_3^{-1} n^{-1} \cdot \text{Ex: } (1100011)_{2}, \\ = 10 \text{m.} (2)_{3} 112^2 + 21 n^2 4 0 n^2 4 0 n^2 3 0 n^2 2^3 1 n^2 2^3 1 (-64 + 32 + 0 + 0 + 0 + 2 + 1)_{30}. \\ \log_{3} x n^{2} (29)_{30}.$	2→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. Depois do ponto decimal os expoentes da base naturalmente prosseguem sende gardos como dígitos negativos d <sub>e.</sub> "2 <sup>2*</sup> *d <sub>e.</sub> " *2 <sup>**</sup> * * + 4,7 *2* *d <sub>e.</sub> " *2 <sup>**</sup> *d <sub>e.</sub> *
2->16	A partir da direlta, converter cada 4 bits para o digito hexadecimal equivalente: $0000 \rightarrow 0,0001 \rightarrow 1,,$ $1001 \rightarrow 9,1010 \rightarrow 9,1010 \rightarrow 9,1010 \rightarrow 9,1010 \rightarrow 1,1011 \rightarrow 1,10$	10 -> Complemento de 1 ou 1's	Se negativo, guardar o sinal, assumir que o número é positivo e converter como para N, mantendo o tamanho da representação. Se positivo, o resultado estará pronto. Se negativo, aplicar a operação de troca de sinal (inverter todos os bits), o que deve gera re ossulado. Ex. (57.3% cl) <sub>17.4</sub> esta p. 12/2-22 esta 10, 12/2-14 esta 0, 12/2-31 esta 0, 12/	2→16	Separar as partes antes e depois da virgula. Para a parte à esquerda da virgula, a partir da direita, converter cada à bits para o digito hexadecimal equivalente: 0000 > 0, 0,001 > 1,, 0100 > 9, 0100 - 4,, 1111 > F. Para a parte à direita da virgula faça o mesmo a partir da esquerda. Ex: (110011,101) <sub>2</sub> =(x) <sub>1x</sub> . 11 0011 , 101(0). Logo, x=(33,A) <sub>1x</sub> .
2→8	A partir da direita, converter cada 3 bits para o digito ctal equivalente: 000 > 0, 001 > 1, 010 - 2,, 110 - 5, 6, 111 > 7. Ex: (10011101) <sub>1</sub> =(x) <sub>1</sub> , 10 011 101. Logo, x=(235) <sub>1</sub> .	Complemento de 1 ou 1's> 10	Se negativo (quando último bit da representação *1), aplica a operação de troca de sinal (inverte todos or bits) e converte como na conventão n -> 10. No final, acrescenta o sinal. Ex (±1001100); ····a tau="10, ··· Como o número é negativo, obtém-se o positivo correspondente entes de converter, o que gera 00110011. Agora converte-se:  0°2'4°0'2'4*12'3'12'4'0°2'4'0°2'4'12'11'2'=(0+0+32+16+0+0+2+1) <sub>30</sub> . Logo, x=(-51) <sub>30</sub> .	2→8	Separar as partes antes e depois da virgula. Para a parte à esquerda da virgula, a partir da direita, converter cada à bits para o digito cotal equivalente: 000 > 0, 001 > 1, 101 - 32,, 110 - 56, 111 - 57, Para parte à direita da virgula faça o mesmo a partir da esquerda. Ex: (1101,1101) <sub>2</sub> =(x) <sub>2</sub> . 1 101, 110 1(00). Logo, xe(15,64) <sub>3</sub> .
8> 2	Converter cade digito actal em exatamente três bits: $0 > 000, 1 > 001, 2 > 010,, 6 > 110 e? > 111. &: (247)_{10}(x)_{2}. 10 100 111. logo, x=(10100111)_{2}.$	10 -> Sinal Magnitude ou SM	Converter como para N, mantendo o tamanho da representação. Se positivo, o resultado estará pronto. Se negativo, colocar o bit mais à esquerda em 1. Ex: [-43] <sub>22</sub> re[.34] <sub>24-10</sub> a resto 1, 20/2-5 resto 0, 5/2-2 resto 1, 27/2-1 resto 1, 27/2-1 resto 1, 27/2-1 resto 1, 27/2-0, resto 1.00 angle, o resultado intermediário em 8 bits 6 00110101. Como o número é negativo, coloca-se o bit de sinal em 1 assim x=(10110101) <sub>tot em 1 bior</sub>	8→2	Converter cada digito octal em exatamente très bits: $0 > 000, 1 > 001, 2 > 010,, 6 > 110 e 7 > 111. Ex: (247,54)_a=(x), 010 100 111, 101 100. logo, x=(10100111,1011)_s$
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>e</sub> =(x) <sub>10</sub> .  1 100 111 011 = 11 0011 1011. Logo, x=(338) <sub>16</sub> .		Ignorar o bit mais à esquerda e converter os bits restantes como na conversão n -> 10. No final, acrescenta-se o sinal. Ex: (101110110) <sub>ther n trail</sub> (x) <sub>12</sub> , Ignora-se o bits mais à esquerda e converte-se o restante para decimal: 0°2'a:12'2+12'40'2'3+12'40'2'3+12'40'2'40'48+24:16:0442+0 = 118. Como o bit mais à esquerda era 1, o número é negativo, e tem-se que x=(-118) <sub>12</sub> .	8->16	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>a</sub> =(x) <sub>1a</sub> . 10 001 100, 111 010 = 10001100,11101(000). Logo, x=(8C,E8) <sub>1a</sub> .
16 -> 2	Converter cada digito hexadecimal em esatamente quatro bisto 3 - 00001,, 8 - 1000, 9 - 1000, 1 - 3 - 1000, 1 - 1110 Er - 1111 Er - 1111 Er - (5C72) <sub>10</sub> =(9) <sub>2</sub> 101 1100 011 0010. Logo, x=[011110010110010].			16->2	Converter cada dígito hexadecimal em exatamente quatro bits: $0 > 0000, 1 > 0001,, 8 > 1000, 9 > 1001, 0 > 5010,, \varepsilon > 1110 \ e^{\varepsilon} > 1111. \ Er \{SC,72\}_{11}e^{\varepsilon} v _{2}. \ 0101 \ 1100, 0111 \ 0010. \ logo, xe[1011100,0111001]_{2}.$
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 (837) <sub>1/2</sub> +(4) <sub>2</sub> 3.000 0001 0111 = 100 000 010 111. Logo, x=(4027) <sub>3</sub> .			16> 8	Converter da base hexadecimal para base 2 como em 16 $\rightarrow$ 2 (Q+). Depois, reagrupar os bits gerados de 3 em 3 a partir da direita, convertendo cada um destes como na conversão 2 $\rightarrow$ 8. Ex: $(61,7C)_{10}$ $(k)_{10}$ , 0110 0001, 0111 1100 = 1 100 001, 011 111. Logo, $x$ $(144,37)_{10}$ .