m2_SI2025.txt
este arquivo: m2_SI2025.txt
======================================
TEMAS e MATERIAIS PARA ESTUDO - SI2025
(ver também https://pater.web.cip.com.br/MaquInfo/)

Códigos ascii (American Standard Code for Information Interchange) - abaixo os primeiros 128 símbolos ordenados da tabela ascii (que não inclui acentos nem cedilha)

00	000	NUL
01	001	SOH
02	002	STX
03	003	ETX
04	004	EOT
05	005	ENQ
06	006	ACK
07	007	BEL
08	008	BS
09	009	HT
0A	010	LF
0B	011	VT
0C	012	FF
0D	013	CR
0E	014	SO
0F	015	SI
10	016	DLE
11	017	DC1
12	018	DC2
13	019	DC3
14	020	DC4
15	021	NAK
16	022	SYN
17	023	ETB
18	024	CAN
19	025	EM
1A	026	SUB
1B	027	ESC
1C	028	FS
1D	029	GS
1E	030	RS
1F	031	US
20	032	SP
21	033	!
22	034	"
23	035	#
24	036	$
25	037	%
26	038	&
27	039	'
28	040	(
29	041	)
2A	042	*
2B	043	+
2C	044	,
2D	045	-
2E	046	.
2F	047	/
30	048	0
31	049	1
32	050	2
33	051	3
34	052	4
35	053	5

36	054	6
37	055	7
38	056	8
39	057	9
3A	058	:
3B	059	;
3C	060	<
3D	061	=
3E	062	>
3F	063	?
40	064	@
41	065	A
42	066	B
43	067	C
44	068	D
45	069	E
46	070	F
47	071	G
48	072	H
49	073	I
4A	074	J
4B	075	K
4C	076	L
4D	077	M
4E	078	N
4F	079	O
50	080	P
51	081	Q
52	082	R
53	083	S
54	084	T
55	085	U
56	086	V
57	087	W
58	088	X
59	089	Y
5A	090	Z
5B	091	[
5C	092	\
5D	093	]
5E	094	^
5F	095	_
60	096	`
61	097	a
62	098	b
63	099	c
64	100	d
65	101	e
66	102	f
67	103	g
68	104	h
69	105	i
6A	106	j
6B	107	k
6C	108	l
6D	109	m
6E	110	n
6F	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u

76	118	v
77	119	w
78	120	x
79	121	y
7A	122	z
7B	123	{
7C	124	|
7D	125	}
7E	126	~
7F	127	DEL

A primeira coluna, à esquerda, tem a numeração da linha representada em hexadecimal; a coluna no meio, a posição da linha em decimal; e a terceira coluna exibe os símbolos ordenados no padrão ascii.

Assim, o código ascii da letra 'q' minúscula é representado como '113' em decimal ou como '71' em hexadecimal.

Veja que, no decimal 113, o algarismo '1' à esquerda do '13' conta o número de vezes que esgotamos o repertório que vai de 00 a 99 antes de acrescentarmos o '13'.

No caso, esgotamos 1 vez o repertório de 00 a 99 (uma catraca contadora faria isso) e acrescentamos 13. Percorrer 1 vez o repertório 00 a 99 significa contar 100 (cem).

Temos aqui 1 x 100 + 13 = 113.

Ou, visto de outro modo, no mesmo decimal '113', o '11' à esquerda do '3' conta o número de vezes (11 vezes) que esgotamos o repertório de dez dígitos (que vai de 0 a 9) antes de acrescentar o '3'

Temos aqui 11 x 10 + 3 = 113.

Isso é, claro, bem conhecido de todos nós.

De modo similar, em hexadecimal, o algarismo à esquerda "conta" o número de vezes que esgotamos o repertório de algarismos à direita.

No exemplo do hexadecimal '71', o '7' à esquerda conta o número de vezes que esgotamos o repertório à direita (16 dígitos, de 0 a F, isto é, de zero a quinze).

Temos, então, 7 x 16 + 1 = 113.

Portanto, '71' em hexadecimal representa o mesmo número (ou posição da linha na tabela) que '113' em decimal.

Em decimal, com dois dígitos, podemos representar (10 x 10 = 100) números entre 00 e 99 (9 é o "último" algarismo do repertório decimal).

Em hexadecimal, com dois dígitos, podemos representar (16 x 16 = 256) números entre 0 e 255, isto é, entre 00 e FF (F é o "último" algarismo do repertório hexadecimal)

Portanto, para numerar as 128 linhas da primeira parte da tabela ascii bastam dois dígitos hexadecimais.

A cada símbolo representado na tabela ascii acima (coluna à direita) letra ou dígito corresponde um par de algarismos hexadecimais (1 byte = 8 bits) representando a posição do símbolo na tabela, aqui também numerada em decimal

O código ascii de um símbolo é a posição (número de linha) do símbolo na tabela. Podemos representar a posição de um símbolo em qualquer sistema de numeração,

mas isto não muda a posição do símbolo (a linha da tabela onde ele se encontra)
e, portanto, não altera o seu código ascii.

Assim como 'III', 'três' e '3' representam o mesmo número, indicar em decimal,
em hexadecimal, ou em outra forma qualquer uma posição na tabela ascii é uma
referência para o mesmo símbolo, para a mesma linha da tabela. Trata-se do mesmo
código ascii escrito de formas diferentes.

======================================

Abaixo, sob a forma de tabela de 16 colunas, está a representação hexadecimal da
sequência de códigos ascii das 2 primeiras linhas deste arquivo, a saber

======================================
TEMAS e MATERIAIS PARA ESTUDO - SI2025

```
3d 3d 3d 3d 3d 3d 3d 3d  3d 3d 3d 3d 3d 3d 3d 3d
3d 3d 3d 3d 3d 3d 3d 3d  3d 3d 3d 3d 3d 3d 3d 3d
3d 3d 3d 3d 3d 3d 0a 54  45 4d 41 53 20 45 20 4d
41 54 45 52 49 41 49 53  20 50 41 52 41 20 45 53
54 55 44 4f 20 2d 20 53  49 32 30 32 35 0a
```

Foram escolhidas linhas sem letras acentuadas para simplificar esta análise
preliminar e manter os símbolos entre os 128 da primeira parte da tabela ascii.

Na tabela temos, em sequència
38 códigos 3d
1 código 0a
38 códigos diversos
1 código 0a
51 códigos diversos
1 código 0a

Não é estranho especular que:

1)o código '3d' representa o símbolo '='

2)o código '0a' representa o 'pula-linha'

3)os demais códigos (abaixo) devem representar as letras e dígitos do texto, um
a um

```
T = 54h (54 hexadecimal)
E = 45h
M = 4dh
A = 41h
S = 53h
  = 20h
e = 65h
  = 20h
M = 4dh
A = 41h
T = 54h
E = 45h
R = 52h
I = 49h
A = 41h
I = 49h
S = 53h
  = 20h
```

P = 50h
A = 41h
R = 52h
A = 41h
  = 20h
E = 45h
S = 53h
T = 54h
U = 55h
D = 44h
O = 4fh
  = 20h
- = 2dh
  = 20h
S = 53h
I = 49h
2 = 32h
0 = 30h
2 = 32h
5 = 35h

A lista (tabela) de correspondências acima é convencional. Ao conjunto de símbolos individuais à esquerda corresponde o conjunto de símbolos (ordenados, ou posições) ascii à direita, aqui representados em hexadecimal com dois dígitos, como vimos.

Outras especulações úteis:

Se 30h corresponde ao algarismo '0', 31h ao algarismo '1' e 32h ao algarismo '2', como listado acima, não é díficil deduzir os outros algarismos da tabela ascii

3 = 33h
4 = 34h
5 = 35h
6 = 36h
7 = 37h
8 = 38h
9 = 39h

É o que verificamos consultando a tabela ascii apresentada antes.

Atenção. 33h representa o algarismo 3, e não o número 3. A representação do número 3 é '3' em hexa e em decimal.

Para ser mais preciso, a codificação original do texto acima é UTF-8 (UnicodeTransformationFormat, 8 bits), uma extensão do ascii mais universal que tem os primeiros 128 (de 0 a 127) códigos de símbolos coincidentes com o ascii.

Sabemos que para representar potências de 10 é cômodo usar um sistema de notação decimal.

Somar múltiplos de 10 a certo padrão implica incrementar a(s) posições(s) decimais correspondentes do padrão, uma espécie de "contador" do número de vezes que esgotamos o repertório à direita do "contador":

[200 + 1345 = (200 + 1300) + 45 = 1500 + 45]; o repertório de 00 a 99 havia sido, no exemplo 1345, esgotado 13 vezes e, com o acréscimo de 200, foi esgotado 15 vezes ([13 + 2 = 15] x 100, e adicionado o deslocamento 45)

Multiplicar por 10 certo padrão implica somente deslocar a posição da vírgula - no caso de inteiros, reduzir ou aumentar o número de zeros à direita do padrão.

Mas, por que usar um sistema hexadecimal para representar códigos de letras, dígitos e outros símbolos?

A representação de informações em um computador digital comum não é decimal nem hexadecimal e sim binária, embora isto não tenha sido sempre assim: os primeiros computadores digitais de programa armazenado em memória eram (bi)quinários, uma mistura de representações nas bases 2 e 5.
(Ver https://en.wikipedia.org/wiki/Bi-quinary_coded_decimal)

Assumindo em binário cada posição elementar (cada "casa") apenas dois estados significativos, ou zero ou um (não dez, como em decimal, nem dezesseis como em hexadecimal) deveria ser mais cômodo indicar os códigos de cada símbolo em notação binária.

E de fato isto é verdade. Contando em binário (assim como contamos em decimal e hexadecimal acima), temos, com quatro posições binárias (para escrever pouco, por enquanto):

```
0000 zero
0001 um
0010 dois
0011 três
0100 quatro
0101 cinco
0110 seis
0111 sete
1000 oito
1001 nove
1010 dez
1011 onze
1100 doze
1101 treze
1110 quatorze
1111 quinze
```

Como temos dezesseis símbolos compostos diferentes, os binários em 4 posições de zero a quinze, podemos fazer corresponder a cada um deles, a título de abreviatura, um algarismo hexadecimal, sem faltar ou sobrar nenhum

```
0  0000 zero
1  0001 um
2  0010 dois
3  0011 três
4  0100 quatro
5  0101 cinco
6  0110 seis
7  0111 sete
8  1000 oito
9  1001 nove
A  1010 dez
B  1011 onze
C  1100 doze
D  1101 treze
E  1110 quatorze
F  1111 quinze
```

A lei de formação do sucessor e do antecessor de qualquer representação binária é equivalente à usada para os decimais e hexadecimais.

Quando esgotamos o repertório elementar em uma posição à direita (isto é, quando voltamos ao zero), acrescentamos uma unidade ao valor na posição imediatamente à esquerda do zero.

Em decimal, por exemplo, para obtermos o sucessor de 129 - acrescentando 1 ao 9 - precisamos indicar a variação da unidade que passou de 9 para zero, assim como a variação da dezena, que passou de 2 para 3.

Assim, dizer que 129 + 1 = 130, ou que 129999 + 1 = 130000, significa dizer que acrescentamos uma unidade à casa das unidades e propagamos o "vai-um" para a posição à esquerda porque o sucessor de 9(999) não é zero, mas 10(000).

Em binário, do mesmo modo, dizer que 1011 + 1 = 1100 significa dizer que acrescentamos uma unidade à casa das unidades e propagamos o "vai-um" para a casa seguinte à esquerda.

```
1011
0001 +

=

 +1
1010

=

+1
1000

=

1100
```

Se tivéssemos 1099 + 1 em decimal, o resultado seria, de modo similar, o decimal 1100, pois esgotamos o repertório 00 a 99 e precisamos "contar" isto na posição à esquerda.

Para representar potências de 2 (operações x 2 sucessivas) é cômodo usar uma notação binária, assim como para representar potências de 10 (operações x 10 sucessivas) são cômodos os decimais.

Como o sucessor de 1111 (quinze) em binário é 10000 (dezesseis), vemos que ao chegarmos a 16 (10000 em binário) este 1 à esquerda dos zeros marca o esgotamento do repertório de arranjos dos quatro binários à direita, a saber, marca a volta ao zero da série vista acima.

Escrevendo com 8 posições binárias a mesma série:

```
0000 0000 zero
0000 0001 um
0000 0010 dois
0000 0011 três
0000 0100 quatro
0000 0101 cinco
0000 0110 seis
0000 0111 sete
0000 1000 oito
0000 1001 nove
0000 1010 dez
0000 1011 onze
0000 1100 doze
0000 1101 treze
0000 1110 quatorze
0000 1111 quinze
```

chegamos ao

```
0001 0000 dezesseis
```

O padrão à esquerda 0001 é o "um" da sequência anterior, de modo que podemos derivar que após o próximo esgotamento do repertório de 4 posições binárias à direita - quando o 0001 0000 chegar a 0001 1111, o sucessor deve ser 0010 0000, e assim por diante

Quando tivermos 1 0000 0000 em binário, esgotamos o repertório que vai de 0000 0000 a 1111 1111 (0 a 15 x 0 a 15, ou seja 16 x 16 = 256 linhas já escritas, da linha zero à linha 255)

Como vimos, cada grupo de 4 bits corresponde a algum número entre 0 e 15 e podemos fazer corresponder uma abreviatura conveniente para os zeros e uns, assim

00	0000 0000
01	0000 0001
02	0000 0010
03	0000 0011
04	0000 0100
05	0000 0101
06	0000 0110
07	0000 0111
08	0000 1000
09	0000 1001
0A	0000 1010
0B	0000 1011
0C	0000 1100
0D	0000 1101
0E	0000 1110
0F	0000 1111
10	0001 0000
11	0001 0001
12	0001 0010
13	0001 0011
14	0001 0100
15	0001 0101
16	0001 0110
17	0001 0111
18	0001 1000
19	0001 1001
1A	0001 1010
1B	0001 1011
1C	0001 1100
1D	0001 1101
1E	0001 1110
1F	0001 1111
20	0010 0000
21	0010 0001
22	0010 0010
23	0010 0011
24	0010 0100
25	0010 0101
26	0010 0110
27	0010 0111
28	0010 1000
29	0010 1001
2A	0010 1010
2B	0010 1011
2C	0010 1100
2D	0010 1101
2E	0010 1110
2F	0010 1111
30	0011 0000
31	0011 0001
32	0011 0010

33	0011 0011
34	0011 0100
35	0011 0101
36	0011 0110
37	0011 0111
38	0011 1000
39	0011 1001
3A	0011 1010
3B	0011 1011
3C	0011 1100
3D	0011 1101
3E	0011 1110
3F	0011 1111
40	0100 0000
41	0100 0001
42	0100 0010
43	0100 0011
44	0100 0100
45	0100 0101
46	0100 0110
47	0100 0111
48	0100 1000
49	0100 1001
4A	0100 1010
4B	0100 1011
4C	0100 1100
4D	0100 1101
4E	0100 1110
4F	0100 1111
50	0101 0000
51	0101 0001
52	0101 0010
53	0101 0011
54	0101 0100
55	0101 0101
56	0101 0110
57	0101 0111
58	0101 1000
59	0101 1001
5A	0101 1010
5B	0101 1011
5C	0101 1100
5D	0101 1101
5E	0101 1110
5F	0101 1111
60	0110 0000
61	0110 0001
62	0110 0010
63	0110 0011
64	0110 0100
65	0110 0101
66	0110 0110
67	0110 0111
68	0110 1000
69	0110 1001
6A	0110 1010
6B	0110 1011
6C	0110 1100
6D	0110 1101
6E	0110 1110
6F	0110 1111
70	0111 0000
71	0111 0001
72	0111 0010

73	0111 0011
74	0111 0100
75	0111 0101
76	0111 0110
77	0111 0111
78	0111 1000
79	0111 1001
7A	0111 1010
7B	0111 1011
7C	0111 1100
7D	0111 1101
7E	0111 1110
7F	0111 1111
80	1000 0000
81	1000 0001
82	1000 0010
83	1000 0011
84	1000 0100
85	1000 0101
86	1000 0110
87	1000 0111
88	1000 1000
89	1000 1001
8A	1000 1010
8B	1000 1011
8C	1000 1100
8D	1000 1101
8E	1000 1110
8F	1000 1111
90	1001 0000
91	1001 0001
92	1001 0010
93	1001 0011
94	1001 0100
95	1001 0101
96	1001 0110
97	1001 0111
98	1001 1000
99	1001 1001
9A	1001 1010
9B	1001 1011
9C	1001 1100
9D	1001 1101
9E	1001 1110
9F	1001 1111
A0	1010 0000
A1	1010 0001
A2	1010 0010
A3	1010 0011
A4	1010 0100
A5	1010 0101
A6	1010 0110
A7	1010 0111
A8	1010 1000
A9	1010 1001
AA	1010 1010
AB	1010 1011
AC	1010 1100
AD	1010 1101
AE	1010 1110
AF	1010 1111
B0	1011 0000
B1	1011 0001
B2	1011 0010

B3	1011 0011
B4	1011 0100
B5	1011 0101
B6	1011 0110
B7	1011 0111
B8	1011 1000
B9	1011 1001
BA	1011 1010
BB	1011 1011
BC	1011 1100
BD	1011 1101
BE	1011 1110
BF	1011 1111
C0	1100 0000
C1	1100 0001
C2	1100 0010
C3	1100 0011
C4	1100 0100
C5	1100 0101
C6	1100 0110
C7	1100 0111
C8	1100 1000
C9	1100 1001
CA	1100 1010
CB	1100 1011
CC	1100 1100
CD	1100 1101
CE	1100 1110
CF	1100 1111
D0	1101 0000
D1	1101 0001
D2	1101 0010
D3	1101 0011
D4	1101 0100
D5	1101 0101
D6	1101 0110
D7	1101 0111
D8	1101 1000
D9	1101 1001
DA	1101 1010
DB	1101 1011
DC	1101 1100
DD	1101 1101
DE	1101 1110
DF	1101 1111
E0	1110 0000
E1	1110 0001
E2	1110 0010
E3	1110 0011
E4	1110 0100
E5	1110 0101
E6	1110 0110
E7	1110 0111
E8	1110 1000
E9	1110 1001
EA	1110 1010
EB	1110 1011
EC	1110 1100
ED	1110 1101
EE	1110 1110
EF	1110 1111
F0	1111 0000
F1	1111 0001
F2	1111 0010

F3	1111 0011
F4	1111 0100
F5	1111 0101
F6	1111 0110
F7	1111 0111
F8	1111 1000
F9	1111 1001
FA	1111 1010
FB	1111 1011
FC	1111 1100
FD	1111 1101
FE	1111 1110
FF	1111 1111

Portanto, a conversão entre hexadecimais e binários é imediata: separando de 4 em 4 as posições binárias, cada grupo de 4 binários terá um algarismo hexadecimal correspondente.
E, de modo inverso, associando a cada algarismo hexadecimal o grupo de 4 bits (o desenho) correspondente, podemos converter facilmente hexas em binários, ou binários em hexas de qualquer tamanho.

Assim, finalmente,

ffef67890 hexa corresponde a 1111 1111 1110 1111 0110 0111 1000 1001 0000 em binário

e 111010110 em binário, isto é, agrupando de 4 em 4 bits da direita para a esquerda e completando com zeros,

0001 1101 0110 corresponde ao hexadecimal 1d6 (ou 1D6)

Para outras bases, em particular as correspondentes a potências de dois, o procedimento é simples e similar.

EXERCÍCIO FÁCIL:
Com as tabelas, representar o seu primeiro nome e o seu último sobrenome, separados por espaço, em ascii, sem acentos ou cedilha, mas usando maiúsculas e minúsculas onde for o caso.

EXERCÍCIO NÃO MUITO DIFÍCIL:
Evitando usar tabelas, representar em ascii (binário, hexa e decimal) os 9 algarismos do seu dre, sabendo que o algarismo zero tem a representação ascii 30h (dois dígitos hexa).

======================================
fim_de_m2_SI2025.txt