ASCII

Review the conversion from one base to another in text as well as two's complement.

Table 1: ASCII (American Standard Code for Information Interchange)

Dec	Hex	Sym	Dec	Hex	Sym	Dec	Hex	Sym	Dec	Hex	Sym
0	0	NUL	32	20		64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(	72	48	H	104	68	h
9	9	TAB	41	29	)	73	49	I	105	69	i
10	A	LF	42	2A	*	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	-	77	4D	M	109	6D	m
14	E	SO	46	2E	.	78	4E	N	110	6E	n
15	F	SI	47	2F	/	79	4F	O	111	6F	o

ASCII

Table 2: ASCII (American Standard Code for Information Interchange)

Dec	Hex	Sym	Dec	Hex	Sym	Dec	Hex	Sym	Dec	Hex	Sym
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	

Assembly Directives

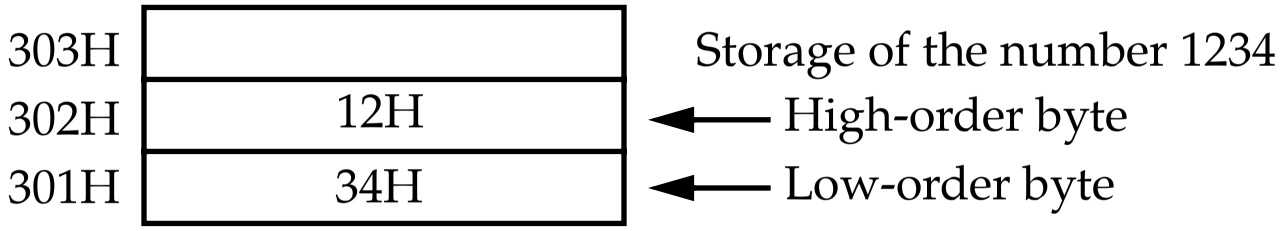
ASCII: Stored using an assembler directive **db**:

```
floatstr db 'Float number -> %f ', 10, 0
main1_str: db '      Rectangular Areas', 10, 0
temp_buf:            times 200 db 0
temp_buf_size:       equ $-temp_buf
```

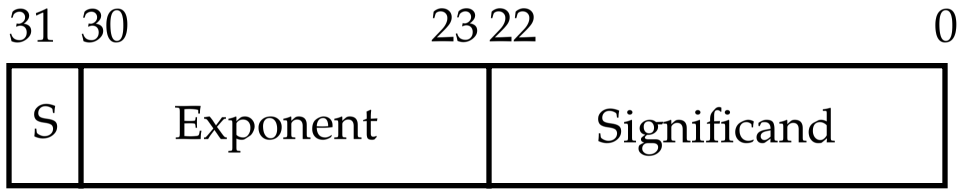
Word-sized (**dw**) and doubleword-sized data (**dd**):

```
neg_exponent:        dd -100
```

Little endian: Least significant byte is always stored in the lowest-numbered memory location.



Floating Point Formats



Single Percision

Double Percision

For single percision, the sign bit + 8-bit exponent + 24-bit mantissa = 33 bits !

The mantissa has a hidden 1 bit in the leftmost position that allows it to be stored as a 23-bit value.

The mantissa is first normalized to be >= 1 and < 2, e.g., 12 in binary is 1100, normalized is 1.1 X 23.

The exponent is also biased by adding 127 (single) or 1023 (double), e.g. the 3 in the previous example is stored as 127 + 3 = 130 (82H).

Floating Point Formats and Directives

Dec	Bin	Normal	Sign	Expon	Mantissa	
+12	1100	1.1×2^3	0	10000010	1000000 00000000	00000000

There are two exceptions:

The number 0.0 is stored as all zeros.

The number infinity is stored as all ones in the exponent and all zeros in the mantissa. (The sign bit is used to indicate + or - infinity.)

Directive is **dd** for single, **dq** for double and **dt** for 10 bytes:

```
dd 1.2
dq 1.e+10
dt 3.14159265358979323846
```