MC140: lecture #10

today's topic:

characters

strings

characters: storage.

- 1 byte = 8 bits
- numeric values from 0 to 127
- used as indexes into a table

the ASCII table

- ASCII = American Standard Code for Information Interchange
- contains numbers, letters (upper and lower case), escape sequences, punctuation marks...

ASCII.

Oct	Dec	Hex	Char	Oct	Dec	Hex	Char
000	0	00	NUL '\0'	100	64	40	@
001	1	01	SOH	101	65	41	A
002	2	02	STX	102	66	42	B
003	3	03	ETX	103	67	43	C
004	4	04	EOT	104	68	44	D
005	5	05	ENQ	105	69	45	E
006	6	06	ACK	106	70	46	F
007	7	07	BEL '\a'	107	71	47	G
010	8	08	BS '\b'	110	72	48	H
011	9	09	HT '\t'	111	73	49	I
012	10	0A	LF '\n'	112	74	4A	J
013	11	0B	VT '\v'	113	75	4B	K
014	12	0C	FF '\f'	114	76	4C	L
015	13	0D	CR '\r'	115	77	4D	M
016	14	0E	SO	116	78	4E	N
017	15	0F	SI	117	79	4F	O
020	16	10	DLE	120	80	50	P
021	17	11	DC1	121	81	51	Q
022	18	12	DC2	122	82	52	R
023	19	13	DC3	123	83	53	S
024	20	14	DC4	124	84	54	T
025	21	15	NAK	125	85	55	U
026	22	16	SYN	126	86	56	V
027	23	17	ETB	127	87	57	W
030	24	18	CAN	130	88	58	X
031	25	19	EM	131	89	59	Y
032	26	1A	SUB	132	90	5A	Z
033	27	1B	ESC	133	91	5B	[
034	28	1C	FS	134	92	5C	\ '\\'
035	29	1D	GS	135	93	5D	]
036	30	1E	RS	136	94	5E	^

Oct	Dec	Hex	Char	Oct	Dec	Hex	Char
037	31	1F	US	137	95	5F	_
040	32	20	SPACE	140	96	60	`
041	33	21	!	141	97	61	a
042	34	22	"	142	98	62	b
043	35	23	#	143	99	63	c
044	36	24	$	144	100	64	d
045	37	25	%	145	101	65	e
046	38	26	&	146	102	66	f
047	39	27	'	147	103	67	g
050	40	28	(	150	104	68	h
051	41	29	)	151	105	69	i
052	42	2A	*	152	106	6A	j
053	43	2B	+	153	107	6B	k
054	44	2C	,	154	108	6C	l
055	45	2D	-	155	109	6D	m
056	46	2E	.	156	110	6E	n
057	47	2F	/	157	111	6F	o
060	48	30	0	160	112	70	p
061	49	31	1	161	113	71	q
062	50	32	2	162	114	72	r
063	51	33	3	163	115	73	s
064	52	34	4	164	116	74	t
065	53	35	5	165	117	75	u
066	54	36	6	166	118	76	v
067	55	37	7	167	119	77	w
070	56	38	8	170	120	78	x
071	57	39	9	171	121	79	y
072	58	3A	:	172	122	7A	z
073	59	3B	;	173	123	7B	{
074	60	3C	<	174	124	7C	|
075	61	3D	=	175	125	7D	}
076	62	3E	>	176	126	7E	~
077	63	3F	?	177	127	7F	DEL

characters: representation.

- character constants are single characters, surrounded by single quotes
- examples:

  `'A', 'b', '+', '2'`

- data type: **char**
- example:

```
char c = 'A';
```


characters: printing.

- format sequence: %c
- examples:

```
printf( "%c\n", 'A' );
printf( "%c %c\n", 'A', 'E' );
printf( "%c %d\n", 'A', 365 );
```

- output:

```
A
A E
A 365
```


characters: interpretation.

- a numeric value is stored, and this value is *interpreted* as a character
- example:

```
int x = 97;
char c = 'a';
printf( "x = %d %c\n", x, x );
printf( "c = %d %c\n", c, c );
```

- output:

```
97 a
97 a
```

printing a row of characters.

```
#include <stdio.h>
int main( void ) {
  int i;
  char c = '*';
  for ( i=1; i<=10; i++ ) {
     printf( "%c",c );
  } /* end of for */
  printf( "\n" );
} /* end of main() */
```


printing 2 rows of characters.

```
#include <stdio.h>
int main( void ) {
  int i, j;
  char c = '*';
  for ( j=1; j<=2; j++ ) {
     for ( i=1; i<=10; i++ ) {
          printf( "%c",c );
     } /* end of for i */
     printf( "\n" );
  } /* end of for j */
} /* end of main() */
```


nested for loops.

```
for ( j=1; j<=2; j++ ) {
  for ( i=1; i<=10; i++ ) {
     printf( "%c",c );
  } /* end of for i */
  printf( "\n" );
} /* end of for j */
```

inner loop

outer loop

strings.

- storing multiple characters in a single variable
- data type is still char
- BUT it has a *length*
- last character is *terminator* : '\0'
- string constants are surrounded by *double* quotes
- example:

```
char s[6] = "ABCDE";
```


strings: printing.

- format sequence: %s
- example:

```
#include <stdio.h>
int main( void ) {
 char str[6] = "ABCDE";
 printf( "str = %s\n", str );
} /* end of main() */
```

- output:

ABCDE

reading.

- material covered today:
  - DD: 8.1-8.2, appendix D
- *assignment #4 is out today*
- *assignment #4 is due Monday 26 Feb*
- *exam #1 is back today*