MC140: lecture #13

today's topic:

ctype library

arrays

The tendency of people to focus on the meaning of sentences influences their ability to notice some of the obvious features.

---

ctype library.

- character handling library

  `#include <ctype.h>`
- functions include:
  - digit recognition (bases 10 and 16)
  - alphabetic/numeric character recognition
  - upper/lower case character recognition/conversion
  - character type recognition

---

ctype library:
**digit recognition
(bases 10 and 16).**

```
int isdigit( int c );
int isxdigit( int c );
```

---

ctype library:
**alphabetic/numeric
character recognition.**

```
int isalpha( int c );
int isalnum( int c );
```

---

ctype library:
**upper/lower case character
recognition/conversion.**

```
int islower( int c );
int isupper( int c );

int tolower( int c );
int toupper( int c );
```

ctype library:
character type recognition.

```
int isspace( int c );
int iscntrl( int c );
int ispunct( int c );
int isprint( int c );
int isgraph( int c );
```

arrays.

- a string is an *array* of characters
- according to WWWebster:
  array, *noun.*
  1a. a regular or imposing grouping or arrangement. ORDER. *lined up in a soldierly array.*
- data structures consisting of related items of the same type

declaring arrays.

- elements
  – the related items
- length
  – number of elements
- index
  – position number of each element
- examples:
  ```
  int x[25];
  char y[1024];
  ```

a picture.

C[0]	-45
C[1]	6
C[2]	0
C[3]	72
C[4]	1543
C[5]	-89
C[6]	0
C[7]	62

declaration:
```
int c[8];
```

printing contents.

```
#include <stdio.h>
int main( void ) {
  int c[8] = { -45, 6, 0, 72, 1543, -89, 0, 62 };
  int i;
  for ( i=0; i<8; i++ ) {
     printf( "element #%d = %d\n", i, c[i] );
  } /* end for i */
} /* end of main() */
```

reading.

- material covered today:
  – DD: 8.3, 6.1-6.4