Readings for week 2, King Chapters 4, 5 and 6 (expressions, simple statements and loops)

Basic C Data Types (King Chapter 7)

% C has a few basic data types, they are:
  - ¶ A character char
  - ¶ An integer int
  - ¶ A floating point number float and double

% I also consider a pointer to be a basic data type - a pointer is always the size of a machine address and is treated like an unsigned integer

% There are **three modifiers** that are commonly applied to some basic data types
% **long,** which basically doubles the space allocated to integer (int) and float/double data types.
% **short**, which handles ints up to 32768 (it is less important now that computers have more memory).
% **unsigned,** which is applied to integers and changes their range from [-2^31,2^31] to [0,2^32]. Basically, an unsigned value does not interpret its left-most bit as a sign. It is used as part of the value. In other words unsigned values are always positive and so are useful in bit-manipulation operations

See King Page 111, for the range of integer values for 32-bit computers.

Use sizeof operator to determine space needs of datatypes.

King P. 489 and 494 give i/o conversion specifier

Declaration	Space	scanf	printf
char c;	1 byte	%c	%c
short int p;	2 bytes (16 bits)	%hd	%hd
int q;	4 bytes	%d	%d
unsigned int r;	4 bytes	%u	%u
long int s;	4 bytes, (but want 8)		%ld
float u;	4 bytes	%f	%f
double v;	8 bytes		%f
long double x;	8 bytes, but (want 16)		%le

Variables can also be initialized at declaration time:

```
char    c = 'd';
int     q = 4;
float   u = 13.5;
double  v = 7.9e-2;
```

% Thus a declaration has the following general format:
Type VariableName = InitialValue ;

% The **= InitialValue** part of the declaration is optional

% A pointer is declared in the following way
type* variable_name ;

% The type specifies the kind of item that is pointed to

Character Constants

% Character constants are enclosed in single quotes, for example, ‘a’, ‘b’, ‘c’.
There are a number of special characters that must be “escaped” so that they can be recognized. See King P. 119. The common ones are:

New Line	‘\n’
Tab	‘\t’
Backslash	‘\\’

the \ is also used to provide values for octal and hexadecimal numbers. See King page 111.

String Constants (literals)

Literal strings are seen first in print statements, and are enclosed in double quotes. They may form an output message like

```
printf (“Hello World”);
```

or may appear as the format string for outputting the value of an expression or variable

```
printf (“This is an integer of value fifteen %d”, 3*5);
```

We will deal with string variables later after we cove arrays and pointers. See also King, Chapter 13.

Arrays

% Arrays and pointers are very closely related in C
% C basically supports one dimensional arrays, the first subscript value is always 0
% An array is declared in the following way:

```
type ArrayVariable [ size ] ;
```

% The type is the type of the individual array elements and size is the length of the array

For example

```
int  a[100];
```

will produce an array that has 100 integers in it, the first element is a[0] and the last element is a[99]
% Arrays can be initialized at with a list of values
% For example

```
int a[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

% A text string is a one dimensional array of char elements
% For example, a text string can be declared in the following way:

```
char string[25] = "This uses 19 bytes";     /* Note */
```

% A character string constant is enclosed in double quotes ("), for example "this is a text string" - there is a major difference between single and double quotes
% In C and Unix a text string is terminated by a zero byte, this is written as '\0', an empty string requires one byte of storage, in general a string with n characters requires n+1 bytes of storage. Later in the course you will have to be careful to allocate space for that extra byte