C Programming

To bring in code from the C libraries:

```
      #include <stdio.h>      // standard input and output functions
      #include <math.h>       // math functions

      // execution begins at the main function

      int main ( void ) {
         printf("Welcome to C!\n");
      } // end main
```

```
// example of using printf function for output

int x = 3, y = 0;
double z = 3.564327;
printf("x = %d   y = %d   z = %.2f\n", x, y, z);
```

Arithmetic   +   -   *   / %

Review of modulus
37 % 4   what is the remainder when we divide 37 by 4?
37 % 4 = 1
12 % 9 = 3
0 % 7 = 0
7 % 0 = div by zero error

Example: Assume you have an integer stored in **num**.   Find the one's place of num and store it in **one**.

```
int num = ???;
int one = num % 10;
```

Example: Remove the rightmost digit (the one's place) and store the value back into num.

```
num = num / 10;  // integer division
```

The value of an assignment statement is the value that was stored in memory.

```
a = 8;   //  the value of this is 8
x = y = k = 0; // "right associative", these assignments are done
               // from right to left
```

Example: Prime numbers
A prime number has two factors, 1 and itself.
Is 1 prime?   No

Write a function that will return true (non-zero) if its parameter is a prime number.   Otherwise return false (zero).

In C, boolean true is any non-zero value. In C, boolean false is zero.

```
int isPrime ( int num ) {
   if (num < 2)
      return 0;
   int i;
   for (i = 2; i <= sqrt(num); i++) {
      if (num % i == 0)
         return 0;  // not prime
   } // end for
   return 1;  // prime
} // end function
```

Write a main function that will print the first 1000 prime numbers, 10 per line, in nice neat columns.

```
int main (void) {
   int p = 2, count = 0;
   while (count < 1000) {
      if (isPrime(p)) {
         printf("%6d ", p);
         count++;
         if (count % 10 == 0)
            printf("\n");
      } // end if isPrime
      p++;
   } // end while
} // end main
```

The file name in C doesn't have anything to do with the contents of the program. Let's call this prime.c

Now, to compile the program in the Linux terminal window (command line) you would type:

```
      gcc -lm prime.c -o prime
```

And to run the program, type the command:

```
      ./prime
```

. stands for the current directory (folder)
./prime means the file named "prime" located in the current folder

.. stands for the directory one level up

cd ..   changes directory to one level up

cd ../../..   changes to the directory three levels up

cd   changes to your home directory

cd ~   changes to your home directory

Let's look at the compiler command:

```
      gcc -lm primes.c   -o  primes
```

gcc stands for "Gnu C Compiler"

-lm is the flag to request the math library

-o is placed before the name of the executable file

Chapter 9

printf format specification

Placeholder	Data Type of Output
%d or %i	int
%u	unsigned int
%x	hexadecimal (base 16)
%o	octal (base 8)
%f	float or double
%e	exponential notation
%c	char
%s	string (character array)
%p	pointer (an address)

Escape sequences

\n	newline
\t	tab
\\	backslash
\"	quotation mark

Pointer = address = memory location

```
int x = 10;
```

The address of x is written as &x

scanf is the most common input function.

```
//  input a value for x
printf("Type a number: ");
scanf("%d", &x);
```

The placeholder must match the type of the variable you are inputting. Use %lf to input a double. Use %f to input a float.

Don't use scanf to input a char. There is another function for getting just one character from the input.