Random Numbers

Why do we need random numbers?

Although the goal of a programmer is to have complete control over the program, there are many times where a degree of randomness is useful, or necessary.

One example of the use of a random number would be in the development of a game. Since all operations are mathematical in nature, a random number could be used to represent:

- a random direction taken when entering a game
- a random reward given
- a random obstacle encountered
-

Random numbers are also very useful in simulations. By modeling a system on the computer, random numbers can be used to test the behaviour of the system to ensure it performs in the expected manner.

Random Numbers are NOT Random

Although this is a very contradictory statement, it is true. Random numbers are actually generated mathematically, with a carefully chosen equation which gives a reproducible pseudo-random pattern of numbers. The actual numbers given are dependent on the compiler being used. A BASIC compiler creates random numbers that are decimal numbers between 0.0 and 1.0. C compilers generate random numbers that are integers, the range of which is compiler specific, but is generally at least as large as a short int.

Controlling the Values of Random numbers

Obviously, to effectively use random numbers, you must be able to control both the size and the type of number obtained for a particular generated value. Two approaches that can be used are shown below.

Note: add the statement below ONCE in your program to ensure the numbers really look random

srand((unsigned int) time(NULL));   Used to ‘seed’ the random number generator

Integer Numbers

int aNum = rand()%(high – low + 1) + low;   //high and low are both integer values

Decimal Numbers

float aNum = rand()/(float)MAX_RAND*(high-low) + low;   //high and low are both float values

MAX_RAND is compiler maximum value, which may or may not be defined in this manner