159.240 Random Numbers Tutorial

1. Use srand() and rand() to generate 100 random integers.
Print each on a line by itself. You need to #include stdlib.h for this to work. Call srand(123) at the start of your program, and call rand() inside a loop. Don't call srand() more than once.

2. Change your 100 random integers to random floats.
For every random int you make using rand(), R, compute this: float f = R / (RAND_MAX + 1.0f), and of course, don't forget to use %f to print this instead of %d.

3. Change the seed value manually (123 to something like 324235) and make sure you are getting different random numbers.

4. #include time.h, and change srand(123) to srand(time(NULL));
Run your program again, and you should have a new sequence every time. If you run it twice in one second, you'll get the same sequence!

5. Change your program to generate random numbers between 6.0 and 10.0.
Tip: You need to multiply your floats by 5, then add 6.

6. Copy and paste the following code for a linear congruential generator (LCG) into a new program, and make sure it works.

```
#include <stdio.h>
#include <stdlib.h>

unsigned long long int seed = 123;
unsigned long long int m = (unsigned long long int)2 << 31;
unsigned long long int a = 1103515245;
int main() {

    unsigned long long int next = ((a * seed) + 12345) % m;
    printf("next random from 12345 is %llu\n", next);
}
```

Remember that in this code, an unsigned long long int is basically just a big int, that cannot be negative. `%llu` is used for printing these kinds of numbers.

7. Write a function called myrand, which takes one int argument and returns an int.

```
int myrand(int previousrand) {
}
```

Put the code for computing a new random number in that function, and replace `((a*seed)+12345) % m` with a call to `myrand`.

You have just implemented your own LCG! If you have time left and you want to do more, implement the random number generator on the next page.

```
unsigned int mw, mz;  // must be global

int main() {
  float f;

  // load mw and mz - these two numbers make up the seed
  mw = 35;
  mz = 478;

  f = (float) GetUniform();
  printf("%1.2f ", f);
}

unsigned int GetUint() {
  mz = 36969 * (mz & 65535) + (mz >> 16);
  mw = 18000 * (mw & 65535) + (mw >> 16);
  return (mz << 16) + mw;
}

double GetUniform() {
// returns a double in the open interval (0, 1)
  unsigned int u;
  u = GetUint();
  return (u + 1.0) * 2.328306435454494e-10;
}
```