

CSCI 1120 (Low-Level Computing), Spring 2010

Homework 2

Credit: 20 points.

1 Reading

Be sure you have read, or at least skimmed, the readings for 2/15, linked from the [“Lecture topics and assignments”](#) page¹.

2 Programming Problems

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source by sending mail to bmassing@cs.trinity.edu, with each file as an attachment. Please use a subject line that mentions the course number and the assignment (e.g., “csci 1120 homework 2”). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department’s Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (10 points) Write a C program that generates a sequence of 20 “random” numbers using the function `rand()`, prints them, and then prints how many are even and how many are odd. To use `rand()` you will need an `#include` directive for `stdlib.h` (similar to the one for `stdio.h` in the “hello world” program).

By default `rand()` will generate the same sequence of numbers every time the program runs. If you want it to generate different numbers, you can do that by including the following line before the first call to `rand()`:

```
srand((unsigned) time(NULL));
```

If you do this, you will need an `#include` directive for `time.h`.

Sample output:

```
1804289383
846930886
1681692777
1714636915
1957747793
424238335
719885386
1649760492
596516649
1189641421
1025202362
1350490027
```

¹http://www.cs.trinity.edu/~bmassing/Classes/CS1120_2010spring/HTML/schedule.html

783368690
1102520059
2044897763
1967513926
1365180540
1540383426
304089172
1303455736

10 of the above were even, 10 were odd

2. (10 points) Write a C program to compute and print the first N Fibonacci numbers. (Recall the definition of the Fibonacci numbers: $f_0 = 1$, $f_1 = 1$, and for $n > 1$, $f_n = f_{n-1} + f_{n-2}$.) You can hardcode N , but choose a value that seems to you to be both interesting and sensible. (It should be obvious what I mean by “sensible” if you try a large value of N .)

Sample output for $N = 6$ (which is sensible but not very interesting):

```
the 0-th Fibonacci number is 1
the 1-th Fibonacci number is 1
the 2-th Fibonacci number is 2
the 3-th Fibonacci number is 3
the 4-th Fibonacci number is 5
the 5-th Fibonacci number is 8
```