CSCI415—Systems Programming

Spring 2017

Programming Assignment 1

The Drunk Beetle and the Boiling Oil

This assignment requires that you design and implement a small C program. It's primary purpose is simply to reacquaint you with C programming on a Linux system.

A single intoxicated beetle is placed in the exact center of a cardboard square that is suspended over a large vat of boiling oil. The beetle, being quite drunk, behaves rather erratically. It moves exactly 1 inch (in 1 second) in a straight line in a randomly chosen direction. It then passes out for exactly 1 second. When it wakes, it again picks a random direction and moves 1 inch. We are interested in selling insurance to drunk beetles on squares suspended above vats of boiling oil, so we want to compute the average time that a drunk beetle will live after it starts to move for the first time. Clearly, the way to get a handle on this average is to measure the time it takes for a beetle to get boiled for lots of beetles, and then compute the average over the sample. To make the problem a bit easier, assume that the last movement by the beetle lasts a full second. That is, don't worry about taking into account the fact that the last movement of the beetle will neither be a full inch nor a full second.

You should check out the manual pages for:

- `random()`, the "better" random number generator on our system,
- `sin()`, `cos()`, etc., the trigonometric functions,
- `printf()`, the formatted output facility (especially the material about formatted output of floating point numbers), and
- `atoi()`, and related functions that convert strings into numeric quantities.

Note that the trig functions are part of the mathematics library on our systems. This means you have to tell the linker to include that library explicitly. If your source program is called beetle.c, and you want an executable called beetle, do the following:

```
gcc -Wall -o beetle beetle.c -lm
```

A final requirement: your program should be parametric in that two command line argument control your simulation:

- The first argument is a positive nonzero integer which specifies the length of one side of the square in inches. This integer must not underflow nor overflow the machine representation of a long integer (you must enforce this).
- The second argument is a positive nonzero integer which specifies how many beetles are boiled in order to get the average lifetime. This integer must not underflow nor overflow the machine representation of a long integer (again, you must enforce this).

For example, if your binary is called `beetle`, the command line

```
./beetle 20 10000
```

will simulate the actions of 10000 beetles on a 20" by 20" square in order to compute mean beetle lifetime.

Make sure that your program exits gracefully with an error message for "bad" command line arguments. Argument checking is a significant component of this assignment (and the course).

Submission

I will post the details of project submission shortly, but you will be required to submit your source, which the TA will compile and execute on a Linux system in the departmental lab in 121 McGlothlin-Street Hall. Bear this in mind if you do your development on another platform.

Your output for a run should appear exactly like the following line:

```
X by X square, Y beetles, mean beetle lifetime is Z.Z
```

where `X` and `Y` are integers and `Z.Z` is a real (show only a single digit to the right of the decimal point). A concrete example:

```
% ./beetle 10 10000
10 by 10 square, 10000 beetles, mean beetle lifetime is 67.0
%
```

Due

Monday, January 30, at 11:59pm.