CSC112 Fundamentals of Computer Science Spring 2016
Lab 3

The Drunken Sailor: A Computer Simulation

In this lab, we will simulate a drunken sailor staggering around on the poop deck[1] as illustrated in Figure 1.

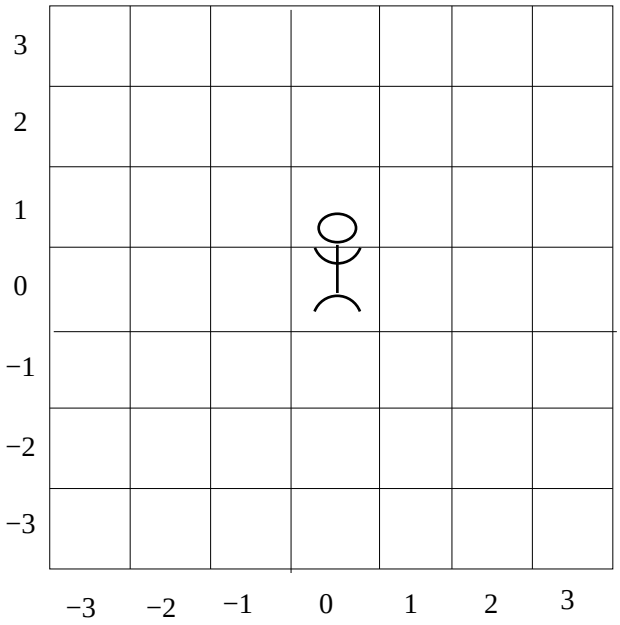

Figure 1: The Poop Deck

Happy Jack and His Dog

At each time step, the sailor either stays in place or moves to one of the eight neighboring squares with equal probability. Eventually, he falls off the poop deck. The idea of the simulation in this lab is to run N experiments. In each experiment, the sailor starts in the middle square, and the experiment runs until he falls off the deck. Your program must count the number of time steps until he falls.

The number of time steps t until our unfortunate sailor falls off the poop deck is a random variable. There are two questions our simulation can answer:

- What is the average (mean) number of time steps until our sailor falls off the deck ?
- What is an estimate of the probability distribution of the number of time steps until our sailor falls off the deck ?

To answer the questions above, write a C++ program to repeat the random walk N times. After each walk, print the number of time steps until the sailor falls. Program requirements include:

1. Use a global constant (const int) to "hard code" the number of experiments $N = 1000000$.
2. Use global constants `MINX`, `MAXX`, `MINY`, and `MAXY` to define the boundaries of the deck.
3. At the beginning of each experiment, the sailor starts at position $(0, 0)$. Use two integer variables to represent the current position.

[1] Anyone know: what is a "poop deck" ?

4. Write a function named `update` to update the current position. The return type should be `void`. Use pass-by-reference to update the two integer variables representing the current position. *Hint:* The expression `rand() % 3` will give you the numbers `0`, `1`, or `2` with equal probability. The change of position in each direction can be computed independently.

5. Write a boolean-valued function named `sailor_falls` to determine if the sailor has reached a position off the deck (and therefore falls).

6. Write a function named `random_walk` to run the random walk until the sailor falls. The function `random_walk` should return an integer: the number of steps until the sailor falls.

7. Write a main function to repeat the random walk N times. For each random walk, output the number of steps until the sailor falls.

Your program will output N positive integers, one per line. With $N = 1000000$, that's a lot of lines of output. Use UNIX re-direction to re-direct the output of your program (stdout) to a file named `nums`. E.g.:

```
gottlieb %  a.out > nums
```

The UNIX shell interprets the `>` character to re-direct the output of the program `a.out` to the file name on the right.

We will use a program named "octave" to draw a histogram for us, and to find three other quantities of interest:

- the average (mean) of the number of time steps until the sailor falls
- the fewest number of steps until the sailor falls
- the largest number of steps until the sailor falls

The octave commands you need are shown below:

```
octave:1> load nums
octave:3> mean(nums)
ans =  << your answer here >>
octave:4> minsteps = min(nums)
minsteps =  << your answer here >>
octave:5> maxsteps = max(nums)
maxsteps =  << your answer here >>
octave:2> hist(nums, maxsteps-minsteps+1 )
octave:6> quit
```

When your histogram appears, use the menus to save it to a file named "histogram.png".

Turn in:

1. Your program source code

2. A PNG graphics file containing your histogram

3. A short write-up including:

(a) The mean number of time steps until the sailor falls

(b) The fewest number of time steps (in any of your experiments) until the sailor falls

(c) The maximum number of time steps (in any of your experiments) until the sailor falls

Save all your work in a directory named "Lab3". Change to your home directory (the parent directory of "Lab3"), and create a file named "lab3.tar" using the command:

```
tar cf lab3.tar Lab3
```

Use `sftp` to upload the file "lab3.tar" to your account on telesto.