

CS 4500; HW3

HW3 assumes that you have successfully accomplished HW1 and HW2. Therefore, we assume that you have a program that does a random walk and collects some information for that walk; and we assume that you know how to make a web based solution that includes graphics.

HW3 requires you to revise and extend your program. You are to do multiple random walks, collect data about these repeated simulations, and then display the results, including a histogram.

When your program starts, you will put up a description of what your program does, and then you will ask the users (via screen prompts that are specific and precise) for the following information:

1. The x and y dimensions of the grid
2. How many random walks the user wants to simulate

You should make sure you get the right kinds of inputs (integers, not fractions or non-digits). Feel free to make reasonable limits on what inputs you will accept, and reprompt if the limits are violated. You should establish a (somewhat arbitrary) limit for the number of steps so that even in theory you will never get an infinite loop.

After gathering those three integers, your program should run the required simulations, and then output to the screen the results of that many simulated walks.

As before, each random walk starts in the lower left, and continues until your marker gets to the upper right or an arbitrary limit is reached. However, after doing one such walk, you will be repeating this process as many times as the user requires. Also, for each piece of information you keep, you will keep a minimum, a maximum, and an average over all simulations. (Number of steps: minimum, maximum, and average; lowest count in the grid: minimum, maximum, and average; and highest count in the grid: minimum, maximum, and average.) In addition, you will keep the information necessary to print out a histogram of how many steps it took to finish the random walks. (If you don't know what a histogram is, and can't figure it out using Wikipedia, just ask.) After all the simulations have been done, display the information in such a way that the user can understand what happened in your simulations. You may keep other information than what I've listed if you think it is interesting.

A steps histogram MUST be included in your output. Other output ideas are up to you. Surely you should include the dimensions of your grid and the number of repetitions. I hope you and your partners will be clever about what things can be shown, and about how to show them effectively.

You have a great deal of latitude in deciding what graphical representation would be most enlightening to viewers. Think creatively about what information you will show to the viewer, and how you will make that information visible. Be careful about screen geography. Part of your grade will be based on whether or not I think your graphics are helpful in understanding what is happening with these multiple random walk simulations.

This is an individual assignment. However, you will be working with a team. You can share code with your team, and no one else. If one of your team members knows graphics better than the others, share. Make sure to include your partners' names in your code's comments. You should also tell how your partners helped you with this assignment. Always give credit where credit is due.

Documentation is central. Each file you submit should have your name, the date, and other helpful information as necessary. An opening comment should make it clear the overall goals of the program, and high level inputs and outputs. Use "paragraphing" comments to break up code. Subprograms

(including each function) should include a comment telling the reader the purpose of the code and any assumptions inherent in the code. Any tricky parts of code should be commented to enhance human readers' understanding of your intent, and the details of the implementation.

Again, your code has to run on the PC in our classroom. Make sure to include instructions so that I can EASILY run your code and see your graphics. Your underlying simulation program has to work, and your graphics have to work in order to receive full credit. Comments with variable declarations are often helpful, and descriptive variable names are essential. Use named constants (or variables that never change) to replace literals wherever practical.

Your program MUST be a web-based program. With a web-based program there is no need for downloading files, linking, compiling, or any such complications. You will be submitting a text file telling me your URL, and an RTF file with your source code.

Be sure to get clarifications from the customer (your instructor..., me...) when necessary.

Keith