CS102, Python, Practical 5, Random Walks.

Please email your solution to Question 1 as an attachment to the course lecturer at the following address:

martin.meere@nuigalway.ie

Please note the following:

(a) **your program should be saved as random_yourfirstname_yourlastname.py;**

(b) **as a backup, put your name and student id number in a comment at the head of the program;**

(c) **the deadline for submission is March the 7th;**

(d) **programs that do not work at all will receive 0%;**

(e) **include sufficient comments for the program to be comprehensible;**

(f) **you may get assistance from the tutors or the lecturer, but people who simply copy or share work risk losing some or all of their marks for this practical.**

1. *A one-dimensional random walk.* On pages 74 to 78 of your notes, a one-dimensional random walk is described. Write a program to simulate m such random walks, each consisting of N steps. The notation used in your notes is used again here without explanation (in other words, you may need to read your notes again). The program should have each of the following features.

   (a) It should take the number of steps in the walk, N, as input.

   (b) It should take the number of times the walk is to be repeated, m, as input. You should use a large value for m when running the program.

   (c) It should use the outcomes for the m walks (with m large) to obtain an estimate for the root mean square distance, $\sqrt{\langle D_N^2 \rangle}$. If D_1, D_2, D_3, ..., D_m are the outcomes for the m walks, then an estimate for $\sqrt{\langle D_N^2 \rangle}$ is given by

$$\sqrt{\frac{D_1^2 + D_2^2 + D_3^2 + \ldots\ldots + D_m^2}{m}}.$$

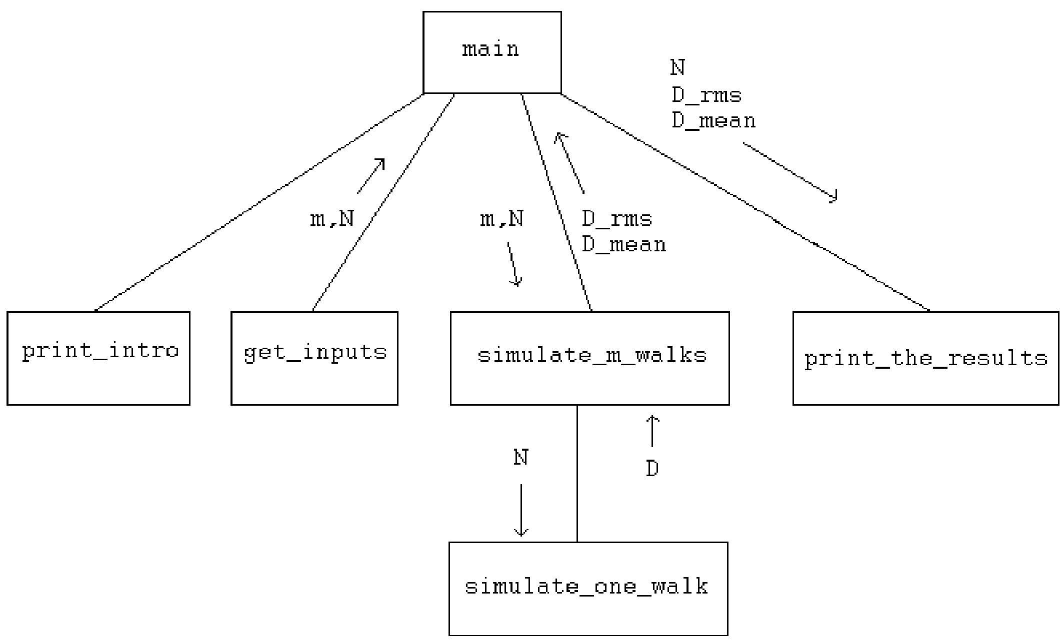
The estimate, and the estimate divided by $\sqrt{N}$, should be printed to the screen.

(d) It should use the outcomes for the m walks (with m very large) to obtain an estimate for the mean distance, $\langle |D_N| \rangle$. If D_1, D_2, D_3, ..., D_m are the outcomes for the m walks, then an estimate for $\langle |D_N| \rangle$ is given by

$$\frac{|D_1| + |D_2| + |D_3| + ... + |D_m|}{m}.$$

The estimate, and the estimate divided by $\sqrt{N}$, should be printed to the screen.

Use your program to estimate $\langle |D_N| \rangle$, $\langle |D_N| \rangle / \sqrt{N}$ for $N = 10^r$ with $r = 1, 2, 3, 4$. Paste in your answers in a comment at the end of your program.



A suggested structure chart for Question 1. In the chart `D_rms` and `D_mean` refer to the estimates for $\sqrt{\langle D_N^2 \rangle}$ and $\langle |D_N| \rangle$, respectively.

2. *The drunken sailor problem.* On pages 78 to 81 of your notes, a two-dimensional random walk is described. Write a program to simulate m such random walks, each consisting of N steps.

   Hint. All you need to do is to copy and paste your program from Question 1 and amend the `simulate_one_walk` function.

3. *Graphical program for a 2D random walk.* You have been emailed (or will be shortly) the program `plot_random.py`. Experiment with this program for various *allowed* values of `r`, where `r` is the length of a step. If you have the patience, try `r = 1`, `N = 100000`.

   Modify this program so that the sailor can move in *any* direction.

   Hint: The modified program should be alot simpler than the program I've given you. Suppose that the sailor's step is still of fixed length `r`, but suppose now that he can move at any angle `theta` relative to the x-axis. Then let (random angle in radians)

$$\texttt{theta} = 2 * \texttt{math.pi} * \texttt{random}()$$

   and then set (polar coordinates)

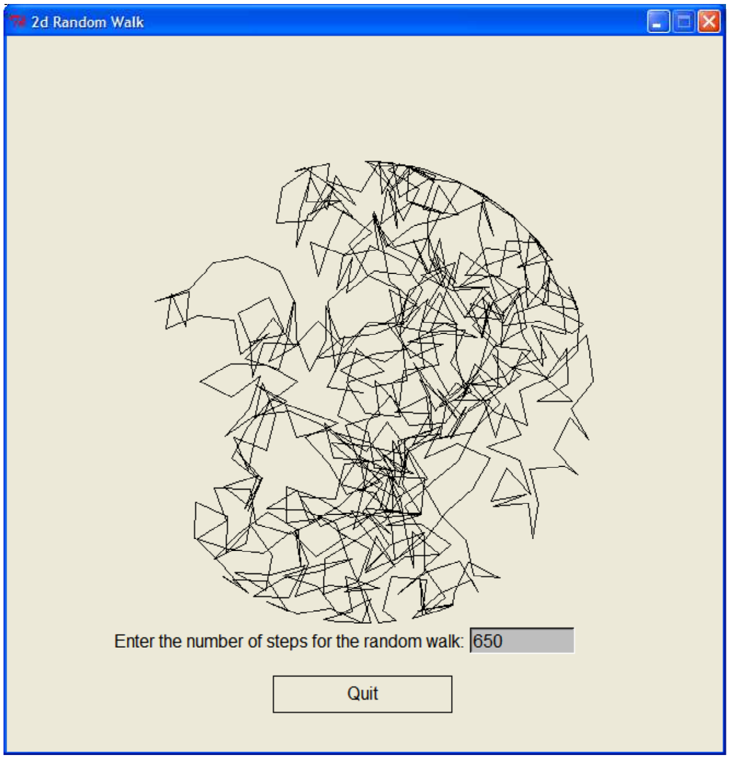
$$\begin{aligned} \texttt{x\_new} &= \texttt{x} + \texttt{r} * \texttt{math.cos}(\texttt{theta}) \\ \texttt{y\_new} &= \texttt{y} + \texttt{r} * \texttt{math.sin}(\texttt{theta}) \end{aligned}$$

   For simplicity, confine the path of the sailor to lie inside a circle of a given radius `R` (try `R = 65`). One way of doing this is to check the value of $\texttt{D} = (\texttt{x\_new} * *2 + \texttt{y\_new} * *2) * *0.5$ every time you calculate a pair `x_new`, `y_new`; if $\texttt{D} > \texttt{R}$, then set

$$\begin{aligned} \texttt{x\_new} &= \texttt{x\_new} * \texttt{R}/\texttt{D} \\ \texttt{y\_new} &= \texttt{y\_new} * \texttt{R}/\texttt{D} \end{aligned}$$

   There are some sample runs of these programs on the next page.

Sample run of the original graphics program with `N = 3000` and `r = 3`.



Sample run of the modified program with `N = 650` and `r = 10`.