HOMEWORK 5

Due: Friday, July 31th 2020, 11:59 PM

Formatting Reminder: The submitted work **MUST** follow the naming convention listed below.

- "**LastName_UID_HW_05_report.pdf**"
- "**LastName_UID_HW_05_code.zip**"
- "**LastName_UID_HW_05_main.m**"

Submit **two** separate files to the CCLE course website: (1) a pdf of your written report and (2) a .zip file containing all of the MATLAB files written for the assignment.

Use a switch statement to call for which problem to solve in your main script, ***DO NOT*** submit two separate m-files for two problems. Remember to use good coding practices by keeping your code organized, choosing suitable variable names, and commenting where applicable. Any of your MATLAB .m files should contain a few comment lines at the top to provide the name of the script, a brief description of the function of the script, and your name and UID. Any submission that does not follow the above-mentioned format will receive SEVERE point deductions!

1. **The Shared Birthday Problem.**
   Write a Monte Carlo simulation to answer the following question: *how many people are required in a group before it is more likely than not that any two of their birthdays occur in the same week?* Conceptually speaking, your algorithm should model starting with an empty group and introducing new people with random birthdays, one at a time, until a pair of birthdays occurs in the same week.

   Assume that a year has 365 days and that all birthdays are equally likely. Out definition of "same week" for this problem will be if the birthdays are separated by less than 7 days in either direction (e.g. birthdays of day 8 and day 2 occur in the same week, while birthdays of day 9 and day 2 do not). If the *N*th person added has the birthday that produces the math, then your answer for that single trial is *N*. In the following example:

group = [124],	No match, add new person
group = [124, 102],	⋮
group = [124, 102, 217],	⋮
group = [124, 102, 217, 26],	⋮
group = [124, 102, 217, 26, 106],	Match found!

The required number of people was 5. Your solution must account for the fact that a birthday near the end of the calendar year and another birthday near the beginning can produce a match if the effective separation is less than 7 days. For example

...	362	363	364	365	1	2	3	4	...

birthdays on day 364 and 2 occur in the same week, while birthdays of day 362 and 4 do not. Repeat this procedure for 1×10^4 trials, storing the *N*-value for each trial in a 1×10^4 element array, and report the median number of the people required using the exact formatting shown below:

```
Median Number of People = 00
```

Where `00` will be replaced with your answer rounded up to the nearest integer. To get a sense for the distribution of *N*-values, plot a histogram using the built-in function `histogram`. Consider what this median indicates: if yu are in a group larger than this, then there is $>$ 50% chance of two people having a birthday in the same week. Does this number seem reasonable to you? Would you expect this value to increase or decrease if we accounted for the fact that not all birth-dates are equally likely?

Note: You are free to use the built-in `median` function, but do not use the functions `find`, `ismember`, etc., in your solution.

2. **Random Walk Collisions**
In lecture, we saw how to model the behavior of a random walker on a 2D grid using a Monte Carlo simulation. In this problem, we will investigate collisions between two of these random walkers. Specifically, we will use our simulation to answer whether a collision is more likely if both walkers are moving or if *only one* walker is moving.

   (a) Start by writing a program to simulate a single random walker on an 11 $\times$ 11 grid. Similar to what we discussed in lecture, let this random walker have an equal p probability of moving to each orthogonally adjacent cell (up, down, left, and right) and a $1-4p$ probability of staying still, as shown below. For this study, set $p = 0.2$. In order to limit the size of the problem, we will restrict the motion to the confines of our 11 $\times$ 11 grid. If the particle tries to move "past" one of the boundaries on its turn, its position does not change (you can think about this as a particle forfeiting its turn or simply bumping into the wall and staying still).

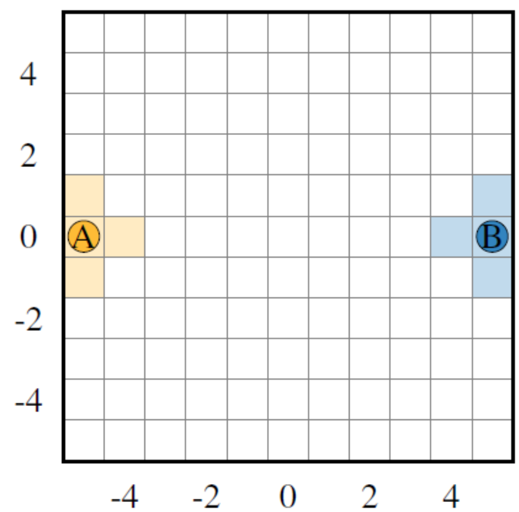

Figure 1: Particles A and B in their initial positions along with a stencil showing potential first moves. Note, your axes do not have to use this exact numbering convention, bu your chosen coordinate system must model an equivalent setup.

   (b) Add another particle to your simulation that moves according to its own randomly generated values. Start particle A at position (-5,0) and particle B at position (5,0) as shown in Figure 1. At each iteration, both particles move simultaneously to an adjacent space using the rules outlined in Part(a). Keep in mind that these random walkers are not following any search strategy, and may blindly swap spaces right past each other. Continue updating the position of both particles until a collision occurs; that is, when both particles occupy the same (x, y) cell. We will only check for a collision once per iteration, after both particles have completed their move. Repeat this procedure for 5,000 trails and report the median number of moves required to produce a collision (you are free to use the built-in `median` function). Report your result using the exact formatting shown below:

```
Median = 00
```

where `00` will be replaced with your answer rounded up to the nearest integer.

Note: In order to reduce computation time, we will enforce a maximum number of possible moves for each trial. If you have simulated 1,000 moves for particles A and B and they have not collided, record 1,000 moves for that trial and proceed to the next one. Even with this safety net, it is possible the full simulation with 5,000 trials will take up to 30 seconds to run. Start with small tests when debugging!

(c) Repeat your simulation from Part(b) except now the position of particle B is fixed at (5,0). Let particle A walk randomly until it collides with B at (5,0). Repeat this procedure for 5,000 trials and report the median number of moves using the same formatting shown above.

(d) If two random walkers become separated, is it faster for both parties to search randomly for one another, or for one to search while one stays put? Is this conclusion dependent on the initial position of both walker (e.g. do you observe the same behavior if they start closer together, A = (-4,0) and B = (4,0), A = (-3,0) and B = (3,0), etc.)? Support your answers with results from your simulation. Are your results in conflict with the commonly-given advice to "stay where you are until help arrives"?

Note: Although you are asked to modify your code to produce results fro your report, please submit the version outlined in Part(b) (both walkers moving) using the initial conditions listed, $p = 0.2$, and 5,000 trials. Do not forget about data abstraction! Just because it is not explicitly required does not mean you cannot clean up your script by packaging ideas as functions. Make sure you include any functions in your .zip file. Also make sure your code creates a time-updated plot with predefined grid spacing and the locations of particles (either as a circle or a square). You may use comments or switch statement for the different parameters for part (c) and (d). Include plots for the initial and final positions for each part in your report.