<u>AI Project 3 Rubric</u>

Name: ________________________________ Score: ____________/22
Date and time received: ____________________ (on time, Thursday, Nov. 17, 2022)
Late penalty (10% per day): _________________

We'll be running your code on our server knuth2. So, make sure your program works on lab machine under Linux environment. YES, you will have to resubmit if it does not (10% per day penalty applies). Your program should consist 2 separate programs (one program/part) so we can grade each part separately.

Part I: We'll run your part I using a command like: **python3 HMMpart1.py stationaryCarReading.csv 20** # the last 2 arguments are the filename for sensor readings and time t.

1. (1pt) Your program reads in the sensor readings from a file. The filename will be passed through **<u>command line argument</u>**. Please see the example **stationaryCarReading10.csv** on the class website for the format of the file.
2. (1pt) The **N** (grid-world size) will be determined by the number in the file. The value of **N** should not be hard coded in your program.
3. (1pt) The **t** (length of the observation) will be determined through **<u>command line argument</u>**. The value of **t** should not be hard coded in your program.
4. (1pt) Your output file follows the required format (see the sample file on class website). When t = 20, name your output file **pMap_atTime20.csv**.
5. (1pt) Output the estimated location of the car (row#, column#) to the console.
6. (6pts) Your implementation works **<u>correctly</u>** with the test cases (probability map and estimated location of the car)
    a. Test case 1 (N = 8, t = 1)
    b. Test case 2 (N = 10, t = 20)
    c. Test case 3 (N = 20, t = 40)

Part II: We'll run your part II using a command like: **python3 HMMpart2.py carReading.csv transitionProb.csv 20** # the last 3 arguments are the filename for sensor readings, filename for transition probabilities, and time t.

1. (1pt) Your program reads in the sensor readings from a file. The filename will be passed through **<u>command line argument</u>** (file format same as **stationaryCarReading10.csv** on the class website).
2. (1pt) The **N** (grid-world size) will be determined by the number in the .csv file. The value of **N** should not be hard coded in your program.
3. (1pt) Your program reads in the transition probabilities from a file. The filename will be passed through **<u>command line argument</u>** (format same as **transitionProb10.csv** on the class website).
4. (1pt) The **t** (length of the observation) will be determined through **<u>command line argument</u>**. The value of **t** should not be hard coded in your program.
5. (1pt) Your output file follows the required format (see the sample file on class website). When t = 20, name your output file **pMap_atTime20.csv**.
6. Output the estimated location of the car (row#, column#) to the console.
7. (6pts) Your implementation works **<u>correctly</u>** with the test cases (probability map and estimated location of the car)
    a. Test case 1 (N = 8, t = 1)
    b. Test case 2 (N = 10, t = 20)
    c. Test case 3 (N = 20, t = 40)

Graduate students/Honor course students:
One-page summary of the research paper. or

Part III.

1. We'll run your part III using a command like: **python3 HMMpart3.py carReading.csv transitionProb.csv 20** # the last 3 arguments are the filename for sensor readings, filename for transition probabilities, and time t.
2. The **t** (length of the observation) will be determined through **<u>command line argument</u>**. The value of **t** should not be hard coded in your program.
3. Output as required (most probable path of the car).
4. Your implementation works **<u>correctly</u>** with the test cases.
    a. Test case 1 (N = 10, t = 20)
    b. Test case 2 (N = 20, t = 30)