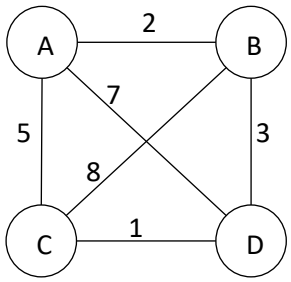
Project 3: Swarms

Goal: To gain experience with simple swarm algorithms.

Description: You'll be implementing the Ant Colony Optimization (ACO) approach to solving the Travelling Salesman Problem (TSP). You'll be provided with a number of sample travelling salesman maps in adjacency matrix format:

	A	B	C	D
A	0	2	5	7
B	2	0	8	3
C	5	8	0	1
D	7	3	1	0



In the example above there 4 cities: A,B,C,D and the length of links between them is specified by the values at the intersection of the corresponding row and column, for example, the link between city A and city B is length 2. Since TSP uses an undirected graph, you will find that the intersection of row A and col B has the same value as the intersection of col A and row B. Note: the values in each row are separated by tabs.

Your program will read in the map from the file, create a set of ants (typically one per city, but if you want to make that tunable, you are free to do so) and then run the ant colony optimization algorithm to try to find the shortest tour.

Parameters: The ACO algorithm uses a number of parameters: alpha, beta, rho, Q, num_iterations, etc. Your code should provide a convenient interface for setting the various parameters. Your readme file should specify what parameter values you found to work best on the provided map files.

Output: While running, your code should periodically print out statistics indicating how the colony is performing: the min, average and max tour lengths, along with what the min tour is in the current iteration. Ideally you should see the min and average tour lengths trending downward as the algorithm progresses – it won't monotonically decrease because the ants choose their next city probabilistically and not purely based on distance or pheromones. After the algorithm has finished running, your code should display the shortest tour from the final iteration as well as the pheromone map in adjacency map format.

Coding Requirements: You are free to choose which language you would like to use in implementing the project. The grader already has C++, Java and Python compilers installed. If you wish to use a different language, you must include a README file with your project submission in which you give step-by-step instructions on where to download the compiler, how to install it (and any libraries that are needed) and how to run your project once the compiler is installed.

Your code is expected to follow all good programming practices, i.e. well-commented, broken into functions, thoroughly tested, etc.

Submission:

Please zip up the entire contents of your project and submit the zip file via Canvas. At a minimum, the zip file should contain:

- Your source code
- Your README file