Report for Project 1

Group 4:
Chaerim Lim
Charlotte Millot
Faheem Abdeen Kulam Magdoon

For project 1, we used Python and did our code on Google Colab to be able to all participate in the project. As requested, we use a weighted A* search algorithm to find the shortest path between two nodes, called in our code **start** and **end**. Those variables are input in our code as the **weight** parameter. We converted the given file as a dataframe and used pandas to access it. It is called **df**. The **reached** list is also created. All of them are the global variables.

The nodes are here represented as dictionary with four keys :

- 'STATE' : which indicates the number of the node
- 'PARENT' : which is the parent node
- 'ACTION' : is the value of the link between the node and its parent
- 'COST' : is the cost from the start to this node

There are 6 functions in our code :

- ASTAR_GRAPH_SEARCH : it is the main function in our code and contain the main loop to go through the graph
- SELECT_BEST_NODE : it compares the nodes in the frontier and their cost calculated with this formula : `w*n['COST']+(1-w)*h`
- HEURISTIC_COST : it calculates the Manhattan distance between two nodes
- EXPAND_GRAPH_SEARCH : it looks at all the children of the node and give a cost which is the cost of the node + the cost of the link in the dataframe
- SEARCH_REACHED : it looks if the node is already reached or not
- PATH_FROM_NODE : it creates the path taken to reach the final node from the start from the end node thanks to the 'PARENT' variable

Our results are in the following table :

Starting node	Ending node	Weight parameter of the A* search	No. nodes generated	Length of the path	Sequence of the nodes on the path
0	19	0.5	27	95	[0, 10, 11, 12, 13, 14, 24, 25, 15, 16, 17, 18, 19]
0	19	0	24	96	[0, 1, 2, 12, 13, 14, 24, 25, 15, 16, 17, 18, 19]
11	97	0.5	35	105	[11, 21, 31, 32, 33, 34, 44, 45, 46, 56, 66, 76, 77, 87, 97]
11	97	0.25	35	105	[11, 21, 31, 32, 33, 34, 44, 45, 46, 56, 66, 76, 77, 87, 97]
40	49	0.5	43	109	[40, 41, 51, 52, 42, 43, 44, 45, 46, 47, 37, 38, 28, 29, 39, 49]
49	40	0.5	38	101	[49, 59, 58, 48, 38, 37, 36, 35, 34, 33, 32, 31, 30, 40]
0	99	0.5	99	No path	
99	0	0.5	37	130	[99, 98, 97, 87, 77, 76, 75, 65, 64, 54, 44, 34, 33, 23, 22, 21, 20, 10, 0]
99	0	0.25	37	130	[99, 98, 97, 87, 77, 76, 75, 65, 64, 54, 44, 34, 33, 23, 22, 21, 20, 10, 0]