Question 1: Compare the three heuristics given in the homework description in terms of number of nodes expanded during the search and analyze your results. Provide a detailed analysis of the performance and discuss general conclusions you can draw regarding the effectiveness of each.

Result:

	Nodes Expanded		
	Euclidean	Manhattan	Chessboard
Input1	19	6	34
Input2	31	19	50
Input3	3	3	5

```
Optimal Path Solution:
Input1: (2,3) (2,4) (2,5) (2,6) (2,7) (2,8)
Input2: (1,3) (1,4) (1,5) (1,6) (2,6)
Input3: (4,4) (4,5) (4,6)
```

Analysis:

Euclidean distance: is the straight-line distance between two points.
Manhattan distance: is the grid-distance between two points.
Chessboard distance: all 8 adjacent cells can be reached in 1 unit.

Looking at the output, the heuristic using the Manhattan Distance provides the optimal solution in terms if number of nodes expanded. The reason is that Manhattan distance is calculated by finding the shortest grid distance between two points. Of the three heuristics, Manhattan distance is the one that matches the actual movements of A and B.

Question 2: Compare the A* search using Manhattan distance with Greedy search using Manhattan distance. For Greedy search, *g(x)=0* in the function *f(x) = g(x) + h(x).* Discuss the performance of each search algorithm in terms of nodes expanded. Are the performances of the two algorithms comparable? Discuss the importance of *g(x)* in performing a search and the effect(if any) it has on search results for the current problem.

Manhattan Search with g(x) = 2:

	Nodes Expanded Manhattan
Input1	6
Input2	19
Input3	3

```
Optimal Path Solution:
Input1: (2,3) (2,4) (2,5) (2,6) (2,7) (2,8)
Input2: (1,3) (1,4) (1,5) (1,6) (2,6)
Input3: (4,4) (4,5) (4,6)
```

Greedy Search with g(x) = 0:

	Nodes Expanded Greedy
Input1	6
Input2	12
Input3	3

```
Optimal Path Solution:
Input1: (2,3) (2,4) (2,5) (2,6) (2,7) (2,8)
Input2: (2,4) (3,4) (2,4) (1,4) (1,5) (1,6) (2,6)
Input3: (4,4) (4,5) (4,6)
```

Comparing the A* search using Manhattan distance vs. Greedy search, the heuristic using Manhattan distance is more optimal in terms of the number of moves needed to make A meet B.

In case of Greedy search, the path cost is zero. Therefore, the Greedy search works more like Best First Search. The algorithm explores the graph by expanding the most promising node chosen according to the Manhattan Distance heuristic.