**Description of task**

You need to find the cheapest solution to the 8-puzzle. Given the initial state and the goal state of 8-puzzle, you need to use BFS, DFS, and Dijkstra algorithms.

For the BFS and DFS algorithms, the cost of moving tile is 1 and ordering of a move is: left, up, right, and down. For example

1 3 4

8 0 2

7 6 5

1 3 4 1 0 4 1 3 4 1 3 4

0 8 2 8 3 4 8 2 0 8 6 2

7 6 5 7 6 5 7 6 5 7 6 5

For the Dijkstra algorithm, suppose that the cost of a move in the 8-puzzle is equal to the number of the moved tile. That is, the cost of moving tile 1 is $1, the cost of moving tile 2 is $2, and so on. Implement the Dijkstra algorithm that finds a cheapest solution; it should read a start state from a file, and print out the shortest path cost and a cheapest sequence of moves that leads to the goal state given in Figure below:



Use the programing language of your choice.

First, you must calculate the number of inversions for the start state (write the codes) to find out if the puzzle is solvable; that is, the program does not have to deal with unsolvable states.

The format for encoding start states is as follows:

<tile> < tile > < tile >

<tile> < tile > < tile >

<tile> < tile > < tile >

Each <tile> is a digit from 0 to 8, where 0 marks the blank space, and the other digits are real tiles. For example, the following file encodes the start state:

7 2 4

5 0 6

8 3 1

Sample file 1:

1 3 4

8 0 2

7 6 5

Solutions: The shortest path cost =11

Sample file 2:

1 3 4

8 0 6

7 5 2

Solution: The shortest path cost = 30