In an enchanted forest, our brave prince finds himself in search for his beautiful princess. The forest itself is in the form of an N by M matrix (N lines and M columns). For every 1x1 patch of forest we know the type of tree that grows there. The prince starts off in the forest patch (start_i, start_j). He also knows the patch where the princess is residing, which is patch (end_i, end_j). During his travel towards the princess, the prince is obliged to stay inside the forest. Also, while he travels, he is sometimes forced to switch between patches of different trees. This is rather costly in terms of money for the prince, each change costing him one gold coin. As he wants to have enough money to buy the princess a nice gift, he would like to minimize the number of coins he is forced to give up for every change in patches. You are to write a program that helps the prince determine the path to the princess that forces the prince to give up a minimum number of gold coins.

- **Input data:**

  The first line of forest.in will contain (in the order presented) the integers N, M, start_i, start_j, end_i, end_j, each seperated by a blank space. The next N lines will each contain M columns of natural numbers representing the tree types for the associated patch.

- **Output data:**

  The output file forest.out should contain a single number representing the minimum number of coins that the prince has to pay during his path.

- **Restrictions:**

  1 <= N, M <= 1000
  1 <= start_i, end_i <= N and 1 <= start_j, end_j <= M
  Points (start_i, start_j) and (end_i, end_j) are never the same

- **Example:**

  forest.in

```
6 5 1 1 5 4
0 0 0 5 6
7 7 1 1 1
1 1 1 3 1
1 1 2 2 1
0 0 9 0 0
0 0 0 0 9
```

  forest.out

```
2
```

  Path found (in bold):

0 0 0 5 6
7 7 **1 1 1**
1 1 1 3 **1**
1 1 2 2 **1**
0 0 9 **0 0**
0 0 0 0 9

  Remark: There could be multiple minimum paths, therefore only the final cost matters

Note:
The solutions should have a Readme file that should contain:
1. a short description of the algorithms you used,
2. the complexity of the algorithms (you must compute it).

The deadline for receiving the homework is December 14th, at 23:59.

Rules for assignments: http://adcfils.wordpress.com/assignements/