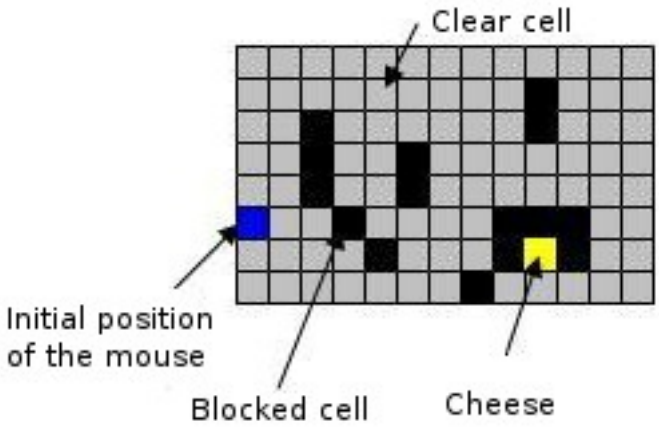
Another Mouse

A) There is a maze of n lines and m columns, a mouse and a cheese, like in the picture below. In the maze there are n*m cells, but some of them are blocked. The mouse wants to get to the cheese on the shortest path possible. He can go in 4 directions(up, down, right and left), but he can`t go through blocked cells. Without using backtracking and shortest path algorithms (Dijkstra, Bellman-Ford...) (that means you must use only what was made in the lab until now), find the shortest path from the mouse to the cheese.



Input data: file “data.in”

```
n m
x y
p q
no_of_blocked_cells
x_bloc1 y_bloc1
x_bloc2 y_bloc2
x_bloc3 y_bloc3
......
x_blocno_of_blocked_cells y_blocno_of_blocked_cells
```

n,m – number of lines/columns
x y – the starting position of the mouse
p q – the coordinates of the cheese
no_of_blocked cells – the number of blocked cells
x_bloc_i y_bloc_i – position of each blocked cell

Output data: file “data1.out”

```
path_length
x1 y1
x2 y2
....
xpath_length ypath_length
```

path_length – the length of the shortest path
x_i y_i – cells from the path

Observations:

The coordinates begin from point (1,1), not from (0,0)

It`s possible that the path doesn`t exist.

There can be more than one shortest path. It`s enough to find one of them.
The length of the path is equal to the number of transitions from one cell to another.

B) To make the life of the mouse more interesting let`s say we bring a cat into the maze. But the cat is lazier and doesn`t want to run after the mouse and goes directly to the strategic cells. A strategic cell is a cell that finds itself on every path the mouse can take to reach the cheese. You must find these strategic points.

Input data: same as in A
Output data: file “data2.out”

```
x1 y1
x2 y2
....
xno_strategic_points yno_strategic_points
```

x_i y_i – the position of every strategic point.

Limitations:
n,m<=1000
Time limit: 10s

Other:
You must put your solution in an archive named *ADC_H<number>_<name>_<group>.zip* and send it to trebedea@gmail.com.
The archive must contain:
your solution
a Readme file with the description of the algorithm and it`s complexity. You should write in it the OS and the compiler you used.
The languages accepted are Java/C/C++/C#/Pascal/Matlab.

The deadline for this assignment is 20 August 2010 23:59.

Don`t forget to read the rules about assignments: http://adcfils.wordpress.com/assignements/