Student Number:		Finish Time:	
Signature:			

University of Cape Town ~ Department of Computer Science
Computer Science 3003S Theory of Algorithms ~ 2014

Procedure

You may consult the electronic Java and C/C++ API documentation (docs.cs.uct.ac.za), and submit to the automatic marker via Vula (vula.uct.ac.za), but nothing else! You may NOT use your class notes, textbooks, internet or files on your flash disk, hard drive, etc.

You may use only the computers in the lab. No tablets, headphones or other personal devices are permitted.

Please sign and return this sheet when you have finished.

Submission

The automatic marker contains a submission entry for each question. Entry and question bear the same name.

Submit your source file within a single compressed, '.ZIP', archive.

Make sure you create a '.ZIP' archive, not a gzipped, '.gz', or tar-gzipped, '.tgz', or other kind of file.

Make sure your source file is the only item within the archive. Especially, avoid submitting an archive containing a folder containing the file.

When submitting a Java source file copied from an editor like Eclipse or Netbeans, please remove any package line that may appear at the beginning of the code.

Practical Test 3 Session 1 Question 2

50 Marks

Each test case that is answered correctly will earn 5 marks.

File names

- Use path.c if you are writing your program in C.
- Use path.cpp if you are writing your program in C++.
- Use Path.java if you are writing your program in Java.

Note that case matters.

Problem Description

Write a program that computes the number of paths that a robot may take when navigating through a given terrain.

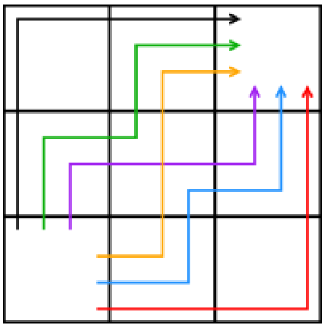
- A terrain is represented by an NxN grid of squares.
- The robot starts in the lower left square of grid, referred to as square (1, 1), and needs to move to the upper right square, referred to as (N, N).
- A terrain also contains K obstacles, each of which blocks a single square on the grid.
- The robot cannot move into a square containing an obstacle, so these reduce the number of possible paths through the grid.
- No two obstacles block the same square, and the starting and ending squares will never contain obstacles.

Question continues on next page

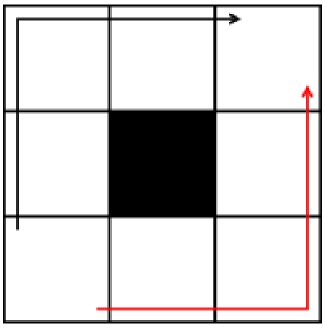
- The robot can only move a single square up or right with each step.

Example:

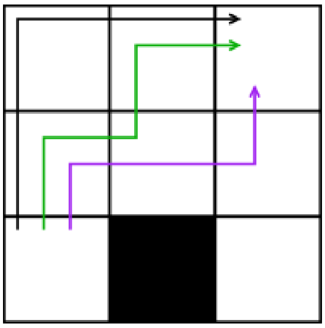
Given a 3x3 grid with no obstacles, there are a total of 6 paths from the bottom left square to the top right, shown below:



Given the same grid with one obstacle on square (2, 2) there are 2 paths:



Given the same grid with one obstacle on square (2, 1) there are 3 paths:



Input and Output

Input consists of a series of lines, each containing up to two integer values.

- The first line of input contains a single integer, N, representing the size of the grid.
- The second line contains a single integer K, representing the number of obstacles on the grid.
- The following K lines of input each contain two integers separated by a space. Each of these lines represents the coordinates on the grid of one of the obstacles.

Output consists of a single integer P (the number of paths from the bottom left grid square, to the top right grid square), followed by a line break --- in Java for example, use `System.out.println`, not `System.out.print`. The automatic marker expects output in this precise form.

Question continues on next page

Constraints:

1 ≤ N ≤ 20
0 ≤ K ≤ N
All obstacles will have coordinates within the dimensions of the grid.
The answer, P, will be bounded by:
0 ≤ P ≤ 1,000,000,000

Sample Input:

```
3
1
2 1
```

Sample output:

```
3
```

END

Question continues on next page