Overview

Author(s)	**Julian Kenwood**	**Keegan Carruthers-Smith**	**Keegan Carruthers-Smith**
Problem	**treetile**	**largerec**	**delivery**
Source	treetile.java treetile.py treetile.c treetile.cpp treetile.pas treetile.hs	largerec.java largerec.py largerec.c largerec.cpp largerec.pas largerec.hs	delivery.java delivery.py delivery.c delivery.cpp delivery.pas delivery.hs
Input file	stdin	stdin	stdin
Output file	stdout	stdout	stdout
Time limit	1 second	1 second	1 second
Number of tests	10	10	10
Points per test	10	10	10
Detailed feedback	Yes	No	No
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Tree Tiles

Julian Kenwood

Introduction

Bruce and Carl have been jet-setting all across the world. They are bored of waiting in airport terminals and decide to play a game which will test the limits of their intellect. The game is called “Tree Tiles”. Although it consists of multiple rounds, we will consider only a single round.

Carl starts off by choosing a subset of the 5 shapes of tiles illustrated in figure 1 and a tree depth, D. Bruce then attempts to count the number of possible ways to tile a complete balanced binary tree of depth D (complete, such that adding another node would always increase the depth) using the tiles that Carl chose. There are an infinite number of each shape of tile available.

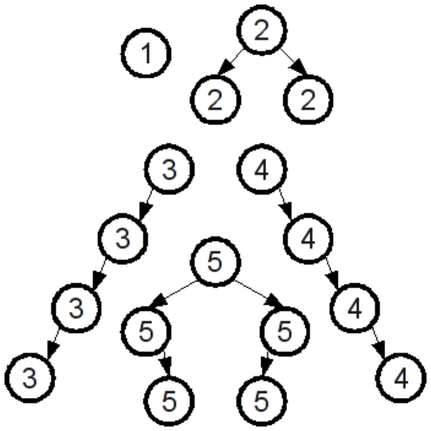

Figure 1: *The five shapes of tiles available. The numbers in the nodes are to number the tiles, with all same-numbered nodes belonging to a single tile.*

Task

Carl recently lost faith in Bruce after he failed to win the Google Code Jam. He has therefore asked you, an innocent bystander, to verify Bruce’s answers.

Example

Assume Carl selects tiles 1 and 2 with $D = 3$. There are five possible tilings:

1. Tile 1 at the root, with tile 2 for both children (illustrated in figure 2.

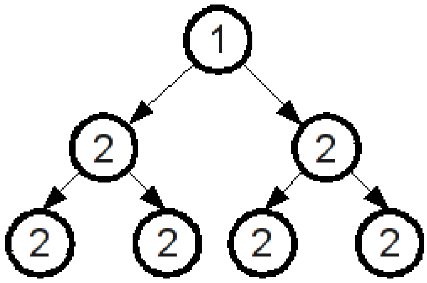

Figure 2: *Example tiling using tiles 1 and 2 on a tree of depth $D = 3$.*

2. Tile 1 at the root, with tile 2 for the left child and tile 1 for the right child, with two further tile 1’s for the children of the right child.

3. Tile 1 at the root, with tile 2 for the right child and tile 1 for the left child, with two further tile 1’s for the children of the left child.

4. Tile 1 at all nodes of the of the tree.

5. Tile 2 at the root, with four tile 1’s on the leaves.

In another example, Carl selects tiles 1, 3, 4 and 5 with $D = 4$. This time there are nine possible tilings.

Input (stdin)

The first line of the input contains a single integer, D: the depth of the tree. Lines 2–6 indicate which tiles Carl chooses. Line $i + 1$ contains a 1 if tile i is chosen, or a 0 otherwise.

Sample input

```
3
1
1
0
0
0
```

Output (stdout)

Output a single integer: the number of possible tilings modulo 10^{80}.

Sample output

```
5
```

Constraints

- $1 \leq D \leq 1\,000$

Additionally, in 50% of the test cases: the number of possible tilings will be at most $2^{63} - 1$.

Time limit

1 second.

Detailed feedback

Detailed feedback is enabled for this problem.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Ben's Mom

Keegan Carruthers-Smith

Introduction

Ben's mother is large and rectangular. So large in fact, we don't know how large she is. She is coming to visit Ben in Cape Town, but we are struggling to find a large enough place for her to sleep. We have found a park, but unfortunately there are trees in the park, which Ben's mother cannot sleep on.

Task

You must find the largest rectangular area in the park, such that no trees are in the rectangle. Note that trees take up the entire space of the cell, and that rectangles must line up with the sides (axis-aligned).

Example

Suppose we have the park such as in the diagram. The black grids represent trees. We can lay out a rectangular area of size 12. The top left corner of the rectangle would be at row 5 column 2, and the bottom right corner is at row 6 column 7.

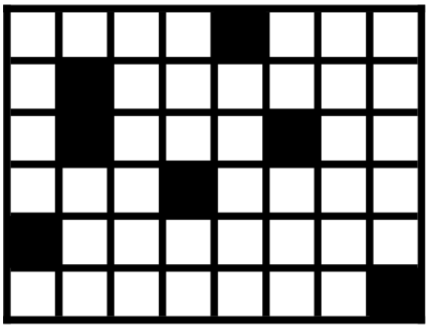

Figure 3: *A layout of the park. Black cells represent trees, empty cells represent places a rectangle can cover.*

Input (stdin)

The first line of input contains two space-separated integers R and C: the number of rows and columns respectively. The next line contains a single integer N: the number of trees in the park. The next N lines contain 2 space separated integers r_n and c_n. This specifies a tree at row r_n and column c_n (indexed from 1).

Sample input

```
6 8
7
1 5
2 2
3 2
3 6
4 4
5 1
6 8
```

Output (stdout)

Sample output

```
12
```

Constraints

- $1 \leq R, C \leq 500$
- $0 \leq N \leq R \times C$
- $1 \leq r_n \leq R$
- $1 \leq c_n \leq C$

Additionally, in 30% of the test cases:

- $1 \leq R \times C \leq 100$

Time limit

1 second.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Newspaper Delivery

Keegan Carruthers-Smith

Introduction

To earn some extra pocket money, you have decided to start delivering newspapers in your local town. As you know, your town only has one-way roads. This makes it is difficult (and sometimes impossible) to deliver newspapers to houses on every road. You are getting paid per road you deliver to. Luckily you are given first choice for the route you are going to deliver on.

Task

Your task is to maximise the number of roads you deliver to along a single route. You may start and end your route from any intersection. You have to obey the rules of the road, so you can only travel in the direction of the one-way street. You want to make as much money as possible, so you are allowed to revisit a road you have delivered to get if it helps you get to a road which has not had a delivery.

Example

In the example input we can deliver to a maximum of 5 roads. We can do this by taking the route: 1 → 5 → 6 → 7 → 5 → 6 → 4. Note that:

- we start at intersection 1 and end at intersection 4;
- there could be more than one way to deliver to 5 roads;
- we visit the road 5 → 6 twice, but only deliver to it once;
- road 2 → 3 and 3 → 2 count as separate roads.

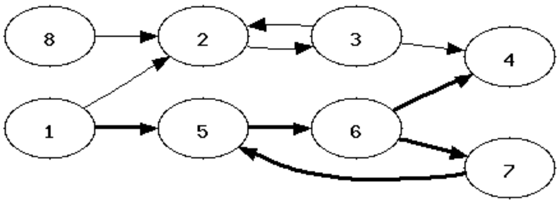

Figure 4: *An image illustrating the sample input. The numbers represent road intersections. The bold arrows indicate roads you deliver to along a maximum route.*

Input (stdin)

The first line of the input contains two space-separated integers, I and R: the number of intersections and the number of roads respectively. The next R lines each contain two space-separated integers, a_i and b_i: this represents a one-way road from intersection a_i to intersection b_i.

Sample input

```
8 10
8 2
1 2
2 3
3 2
3 4
1 5
5 6
6 7
7 5
6 4
```

Output (stdout)

The output contains a single integer: the maximum number of roads you can deliver to.

Sample output

```
5
```

Constraints

- $2 \leq I \leq 1\,000$
- $1 \leq R \leq I \times (I - 1)$
- $a_i \neq b_i$
- There is at most 1 road starting at intersection a and ending at intersection b.

Additionally, in 50% of the test cases:

- $1 \leq I \leq 10$

Time limit

1 second.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.