South African Computer Olympiad
Camp 2, 2008
Day 2

Overview

Author	**Max Rabkin**	**Richard Baxter**	**Marco Gallotta**
Problem	**central**	**tiles**	**declone**
Source	central.java central.py central.c central.cpp central.pas central.hs	tiles.java tiles.py tiles.c tiles.cpp tiles.pas tiles.hs	declone.java declone.py declone.c declone.cpp declone.pas declone.hs
Input file	stdin	stdin	stdin
Output file	stdout	stdout	stdout
Time limit	1 second	1 second	1 second
Number of tests	20	20	20
Points per test	5	5	5
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Grand Central Taxi Rank

Author

Max Rabkin

Introduction

The taxi drivers are tired of long days and little money and want to increase their profits. They hired Carl to streamline their operations but soon after completing the first part of the project (where he minimised the number of routes the taxis drive) he graduated and left for London. He recommended that the taxi drivers hire you for the second part of the project.

The taxi drivers are deciding where they should site the central taxi rank, where all taxis begin their routes.

Task

Given the list of taxi ranks and connections between them, find the one which is most central. The most central means the rank where the distance from it to the furthest rank from it is minimised.

There is exactly one path between any two ranks, and you can assume that taxis stop at each rank for a negligible amount of time. Making this assumption accurate may form a future project.

Example

Consider the taxi ranks as illustrated in Figure 1. If we choose taxi rank 2 to be the central rank then the furthest distance is to rank 4 which takes 11 minutes. This is the best solution.

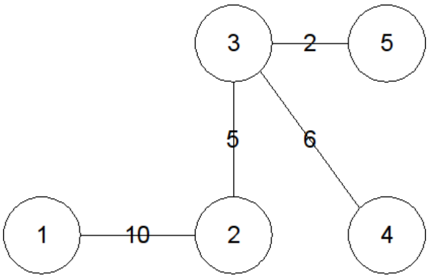

Figure 1: Example of taxi ranks and routes between them.

Input (stdin)

The first line contains an integer, N, the number of taxi ranks (numbered 1 to N). The next $N-1$ lines contain three integers, namely the taxi ranks at each end of a road and the length D_i of that road.

Sample input

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

Output (stdout)

Output a single line containing the number of the most central taxi rank and the distance to the furthest rank from it. If two or more ranks are equally central, output the one with the lowest number.

Sample output

```
2 11
```

Constraints

- $1 \leq N \leq 100\,000$
- $1 \leq D_i \leq 200$

50% constraints

- $1 \leq N \leq 1\,000$

Time limit

1 second.

Scoring

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

Tiling Bruce's Corridor

Author

Richard Baxter

Introduction

As we all know, Bruce is working overseas and earning a good living doing very smart things for big companies. With all this extra income he has gotten himself a new apartment and has decided to retile all the floors. After visiting the tile people he decides on a combination of 2×1 and 3×1 tiles.

He knows exactly what to do for all the rooms but he is a bit concerned about the corridor. His corridor is 2 tiles wide and N tiles long. The problem is that no matter what combination of tiles he tries to use in his corridor, it never seems to be just right. He decides that he will have to try every combination of 2×1 and 3×1 tiles that he can think of.

Task

Given an infinite number of 2×1 and 3×1 tiles, your task is to help Bruce determine how many different ways you can tile a $2 \times N$ corridor using only those tiles.

Example

If you have a 2×3 corridor then there are 4 different ways of laying the tiles as illustrated in the figure below.

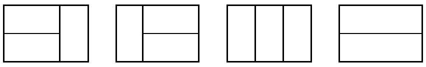

Figure 2: Example showing all configurations of tiling a 2×3 corridor.

Input (stdin)

The first and only line contains a single integer N.

Sample input

```
3
```

Output (stdout)

Output the remainder when the total number of configurations is divided by 1 000 007.

Sample output

```
4
```

Constraints

- $2 \leq N \leq 5\,000\,000$

50% constraints

- $N \leq 5\,000$

Time limit

1 second.

Scoring

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

De-Clone Carl

Author

Marco Gallotta

Introduction

Oh no! Could it be? There are Carls everywhere! He has somehow managed to clone himself on an unimaginable scale. This just goes against all the rules. Bruce was supposed to take over the world, not Carl! If Bruce wakes up and sees this he's not going to be impressed, so we have to solve this quickly.

The South African Army have been contacted and they claim they have a secret weapon that will destroy any clones it hits, while keeping the original being unharmed. Unfortunately they have very limited stock. Fortunately though, the clones are still adjusting to the environment and are therefore easy targets and we can hit many with one shot.

Task

Bruce will wake up in about six hours and it's estimated that it will take an hour to deploy all the weapons. That gives us five hours to come up with a strategy that minimises the number of weapons used. You are trusted with the task of writing a program to calculate this strategy.

The city the clones are taking over is in a $R \times C$ grid pattern. Each Carl clone is situated in a single cell, i.e. no clone is on a cell boundary. A weapon must be launched from the outside of the grid in such a way that it destroys all clones in either a horizontal "row" or vertical "column" of cells as illustrated in Figure 3.

With the guarantee that each weapon fired will destroy all clones in an entire row or column, the strategy you are to devise is which rows and/or columns to select.

Example

To prepare you for the task ahead we provide you with a training exercise. Suppose the city is a tiny size of 4×4 cells and that there are only three clones, in cells $(1, 3)$, $(2, 1)$ and $(4, 3)$.

It would be trivial to destroy all clones with three shots, however, we can do better by noticing that clone one and three (numbered in the order listed above) lie in the same column. We can therefore destroy all clones by shooting column three and row two, using a minimum of two weapons. This strategy is illustrated in Figure 3.

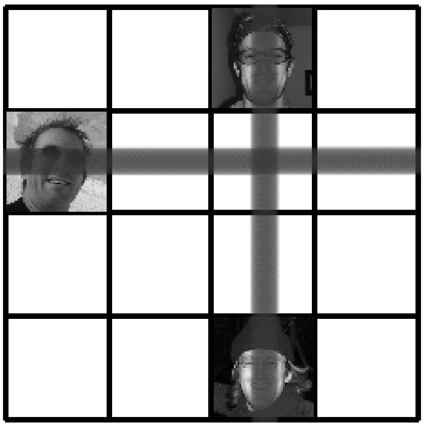

Figure 3: Example with three clones depicted by Carl's face and two weapons fired depicted by the thick lines. Beware the clones' happy looking faces – this is just another evil tactic!

Input (stdin)

The first line of the input contains two space-separated integers R and C, describing the number of rows and columns in the grid. The second line contains a single integer N. The next N lines each contain two space-separated integers, r_i and c_i giving the row and column of the N^{th} clone.

Sample input

```
4 4
3
1 3
2 1
4 3
```

Output (stdout)

The first line of the output must contain the minimum number of weapons required, W. This must be followed by W lines, each describing where to fire a weapon from. This is done by using the format Rn for a weapon firing at row n or Cm for a weapon firing at column m.

If there are multiple strategies using the minimum number of weapons, any of them are acceptable.

Sample output

```
2
R2
C3
```

Constraints

- $1 \leq R, C \leq 500$
- $0 \leq N \leq R \times C$
- $1 \leq r_i \leq R$ for all i
- $1 \leq c_i \leq C$ for all i

35% constraints

- $N \leq 20$

20% constraints

- $1 \leq R, C \leq 10$

Time limit

1 second.

Scoring

A solution that uses more weapons than required will score 0%. A solution that doesn't successfully destroy all clones will score 0%. A solution that is formatted incorrectly, ignoring whitespace, will score 0%. A solution that destroys all clones using the minimum number of weapons will score 100%.