WOBURN CHALLENGE

2018-19 On-Site Finals

Saturday, May 18th, 2019

Junior Division Problems

Automated grading is available for these problems at:
wcipeg.com

For more problems from past contests, visit:
woburnchallenge.com

Problem J1: Conditional Contracts

10 Points / Time Limit: 2.00s / Memory Limit: 16M *Submit online: http://wcipeg.com/problem/wc18fj1*

The cows and monkeys of Scarberia, their long-standing conflicts well behind them at last, have banded together to produce a historical drama documenting their past battles. With the Head Monkey as writer and producer, and the director's chair occupied by the cows' leader, Bo Vine, this will be a collaborative masterpiece to remember!

As part of his preparations, Bo Vine is considering what width of film to use, an important artistic decision. He doesn't want to limit himself to a single width, so he's decided that two copies of the movie will be shot, on two different film widths! Each film width (measured in mm) must be an integer between 1 and 1,000,000,000, inclusive, and the two widths must be distinct.

There are N ($1 \le N \le 100$) acclaimed actors interested in the film, and the Head Monkey and Bo Vine would like to cast as many of them as possible. However, they only work under very specific conditions — the i-th actor insists that they must be captured on film with a width of exactly W_i mm ($1 \le W_i \le 1{,}000{,}000{,}000$). Each actor can be employed if their requirement is satisfied for at least one of the two copies of the movie.

If Bo Vine chooses a pair of distinct film widths optimally, what's the maximum number of different actors who can be employed?

Subtasks

In test cases worth 5/10 of the points, $W_i \le 100$ for each i.

Input Format

The first line of input consists of a single integer, N.
N lines follow, the i-th of which consists of a single integer, W_i, for $i = 1..N$.

Output Format

Output a single integer, the maximum number of actors who can be employed.

Sample Input

```
4
50
70
51
70
```

Sample Output

```
3
```

Sample Explanation

One possibility is using 50mm film for the first copy of the movie and 70mm film for the second copy. This allows the first, second, and fourth actors to be employed. It's impossible for all 4 actors to be employed at once.

Problem J2: Script Doctor

10 Points / Time Limit: 2.00s / Memory Limit: 16M *Submit online: http://wcipeg.com/problem/wc18fj2*

The Head Monkey has just completed her first draft of the film script, and has proudly passed it on to Bo Vine for a readthrough. Unfortunately, Bo has concerns that coarse language in the script might cause the movie to end up with an R rating, which would be a disaster! In order to secure exposure of the movie to younger audiences, he'd like to take a pass over the script himself to ensure that it contains no occurrences of a particularly offensive word: "`bull`".

The script is a non-empty string *S* consisting of at most 1000 characters, each of which is either a lowercase letter ("`a`".."`z`") or an underscore ("`_`"). Bo Vine would like to alter it as little as possible such that, once he's done, the string contains no occurrences of the substring "`bull`". Now, he's not a writer, so he doesn't feel qualified to add anything to the script to accomplish his goal — he'll only remove existing characters from it.

Help Bo Vine remove as few characters as possible from *S* to yield a string with no occurrences of the substring "`bull`". If there are multiple ways to do so, any will do. It's possible that no characters may need to be removed.

Input Format

The first and only line of input consists of a single string, *S*.

Output Format

Output a single string, any maximum-length cut-down version of *S* with no occurrences of "`bull`".

Sample Input

```
the_bull_was_feeling_bulllish
```

Sample Output

```
the_bul_was_feeling_ulllish
```

Sample Explanation

Upon removing two characters (the 7th one, "`l`", and the 22nd one, "`b`"), the resulting string no longer contains "`bull`". It's impossible to achieve this by only removing one character. Note that multiple other outputs with two removed characters would also be accepted.

Problem J3: Behind the Scenes

18 Points / Time Limit: 2.00s / Memory Limit: 16M *Submit online: http://wcipeg.com/problem/wc18fs1*

Today, a full day of filming is set to take place at the monkeys' and cows' joint studios. The studios feature three sound stages, numbered from 1 to 3, and there are initially E_i ($0 \le E_i \le 1000$) pieces of filmmaking equipment on stage i.

A sequence of N ($1 \le N \le 1000$) shots will be filmed, one after another, with the i-th one taking place on stage S_i ($1 \le S_i \le 3$). Whenever a shot is filmed on a certain stage, there must be no equipment present on that stage at the time. Right before each shot, Bo Vine may choose 0 or more pieces of equipment to be moved. Each such piece will be moved from its current stage to a different stage of Bo's choice, at the cost of tipping a stagehand \$1, and will remain there until potentially moved again later.

What's the minimum total cost of stagehand tips which Bo Vine will need to dispense such that each of the N shots will have no equipment present on its stage at filming time?

Input Format

The first line of input consists of three space-separated integers, E_1, E_2, and E_3.
The next line consists of a single integer, N.
N lines follow, the i-th of which consists of a single integer, S_i, for $i = 1..N$.

Output Format

Output a single integer, the minimum total cost of stagehand tips required, in dollars.

Sample Input

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

Sample Output

```
17
```

Sample Explanation

One possible optimal strategy for Bo Vine is as follows:

- Before the 1st shot, move all 4 pieces of equipment from stage 3 to stage 2.
- Before the 2nd shot, move all 6 pieces of equipment from stage 1 to stage 3, and all 7 pieces of equipment from stage 2 to stage 3.

This requires \$4 + \$6 + \$7 = \$17 in stagehand tips.

Problem J4: Top Billing

28 Points / Time Limit: 2.00s / Memory Limit: 16M *Submit online: http://wcipeg.com/problem/wc18fs2*

Two particularly famous actors are starring in the cows' and monkeys' production: Tom Cows and Monkey Freeman. With both of them playing equally important roles, one question is at the forefront of everybody's mind: which of them will receive top billing in the credits and get to be the main face of the movie?

Tom isn't one to stand for playing second fiddle, so he's hatching a plan to discredit Monkey in the eyes of the director, Bo Vine. He's going to sneak into the main studio building in the middle of the night and rearrange its layout, in the hopes of causing Monkey to be late for the following morning's filming session. That'll show him!

The building can be represented as a grid with R rows and C columns ($2 \le C \le R \le 100$). Note that the grid has at least as many rows as it has columns. Tom will choose one starting cell (leading to the parking lot) and one different ending cell (leading to the movie set). These cells may be anywhere in the grid (not necessarily on its edges). For each remaining cell in the grid, he'll choose to either leave it vacant or obstruct it with a wall. He'll make sure that it's possible to reach the ending cell from the starting one through a sequence of horizontally/vertically adjacent vacant cells.

In the morning, Tom and Monkey will begin in the starting cell and will each attempt to make their way to the ending cell, by repeatedly moving from cell to adjacent cell (up, right, down, or left) at a rate of 1 minute per cell. Tom, having arranged the building, will follow a route allowing him to reach the ending cell as quickly as possible. Monkey, on the other hand, will follow the following procedure:

1. Make an ordered list of possible next cells to move to as follows: for each of the directions up/right/down/left (in that order), if the cell adjacent to his current one in that direction is within the grid and doesn't contain a wall, include it in the list. Note that this list will always contain between 1 and 4 cells, inclusive.
2. Move to the cell in the list which is closest to the ending cell by Manhattan distance* (ignoring any walls). If multiple cells in the list have the same minimum Manhattan distance to the ending cell, choose the one of them which appears earliest in the list.
3. If he's arrived at the ending cell, stop. Otherwise, return to Step 1.

* *The Manhattan distance between a cell in (row r_1, column c_1) and another cell in (row r_2, column c_2) is defined as* $|r_1 - r_2| + |c_1 - c_2|$.

For example, consider the following two building layouts (with "`S`" denoting the starting cell, "`E`" denoting the ending cell, and walls marked with "`#`"s):

```
...S
.##.
.##.
E...
```

Monkey would reach the ending cell after 6 minutes (following the sequence of moves ↓↓↓←←←).

```
...S
....
....
E.#.
```

Monkey would walk around forever (following the sequence of moves ↓↓↓↑↓↑↓↑↓…).

Help Tom come up with any building layout which would cause Monkey to successfully arrive the ending cell from the starting one eventually (rather than walking around forever as in the second example above), but at least $2C - 4$ minutes after Tom does. It's guaranteed that at least one such building layout exists for any valid pair of dimensions R and C (with $C \le R$).

Subtasks

In test cases worth 12/28 of the points, $R \le 4$ and $C \le 4$.

Input Format

The first and only line of input consists of two space-separated integers, R and C.

Output Format

Output R lines of C characters each, a grid representing the chosen building layout. Each character must be one of the following:

- "`S`": starting cell (which must appear exactly once)
- "`E`": ending cell (which must appear exactly once)
- "`.`": vacant cell
- "`#`": wall

Sample Input 1

```
4 2
```

Sample Output 1

[Multiple possible accepted grids]

Sample Input 2

```
10 9
```

Sample Output 2

[Multiple possible accepted grids]

Sample Explanation

In the first case, below is an example of a grid which would be accepted:

```
ES
..
.#
#.
```

For this grid, both Tom and Monkey would reach the ending cell in 1 minute, by moving to the left. This is a difference of 0, which is at least as large as the minimum required difference of $2C - 4 = 0$. Therefore, this output would be judged as *correct*.

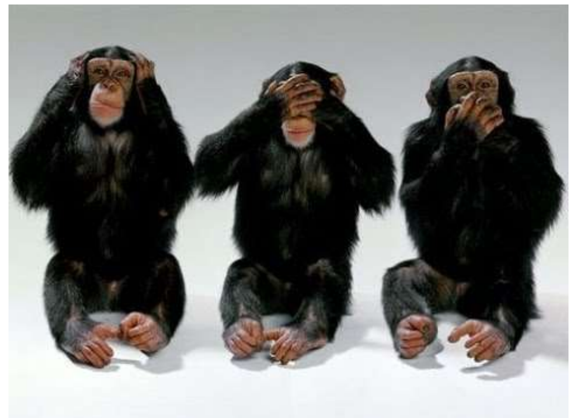
In the second case, below is an example of a grid which satisfies the basic output format requirements, but would *not* be accepted:

```
.........
.........
.........
.........
.........
.........
.........
....#....
.S#...#E.
.........
```

For this grid, Monkey would reach the ending cell in 10 minutes (following the sequence of moves ↑→→↓→→↑→→↓), while Tom would only require 8 minutes (following the sequence of moves ↓→→→→→→↑). This is a difference of 2, which is smaller than the minimum required difference of $2C - 4 = 14$. Therefore, this output would be judged as *incorrect*. Various other grids exist which would be judged as *correct*, however they are not disclosed here.

Problem J5: Screen Time

34 Points / Time Limit: 5.00s / Memory Limit: 64M *Submit online: http://wcipeg.com/problem/wc18fs3*



A trio of rival actors, numbered 1..3 from oldest to youngest, are set to make appearances in the cows' and monkeys' film. However, their contracts state that the oldest of them (actor 1) must receive more screen time than either one of the others! The Head Monkey may need to adjust her script to ensure that that's the case.

There are N ($1 \le N \le 200{,}000$) possible equal-length scenes to include in the script, each of which involves exactly two of the three actors in question. The i-th scene features the two actors A_i and B_i ($1 \le A_i, B_i \le 3$, $A_i \ne B_i$). The Head Monkey may select any non-empty subset of these N scenes to include (note that there are $2^N - 1$ such possible subsets).

Letting S_i be the number of scenes featuring actor i, how many different subsets of scenes could the Head Monkey choose to include such that both $S_1 > S_2$ and $S_1 > S_3$ hold? As there may be many valid subsets, you only need to compute the number of them modulo 1,000,000,007.

Subtasks

In test cases worth 8/34 of the points, $N \le 20$.
In test cases worth another 6/34 of the points, $N \le 200$.
In test cases worth another 6/34 of the points, $N \le 3000$.

Input Format

The first line of input consists of a single integer, N.
N lines follow, the i-th of which consists of two space-separated integers, A_i and B_i, for i = 1..N.

Output Format

Output a single integer, the number of valid subsets of scenes (modulo 1,000,000,007).

Sample Input 1

```
5
3 1
1 2
3 2
2 3
3 1
```

Sample Output 1

```
3
```

Sample Input 2

```
15
3 2
1 2
3 1
2 1
2 1
2 3
1 3
3 2
1 2
1 3
2 3
3 1
1 3
2 3
2 1
```

Sample Output 2

```
7266
```

Sample Explanation

In the first case, if the subset of scenes (1, 2) is chosen, then $S_1 = 2$ while $S_2 = S_3 = 1$, making this a valid subset. Subsets (1, 2, 5) and (2, 5) are also valid. Therefore, the answer is 3 modulo 1,000,000,007 = 3.