South African Computer Olympiad
Final Round
Day 1

Overview

Author	**Keegan Carruthers-Smith**	**Harry Wiggins**	**Max Rabkin**
Problem	**lprefix**	**knights**	**trojan**
Source	lprefix.java lprefix.py lprefix.c lprefix.cpp lprefix.pas	knights.java knights.py knights.c knights.cpp knights.pas	trojan.java trojan.py trojan.c trojan.cpp trojan.pas
Input file	lprefix.in	knights.in	trojan.in
Output file	lprefix.out	knights.out	trojan.out
Time limit	1 second	1 second	2 seconds
Number of tests	10	10	10
Points per test	10	10	10
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Longest Prefix

Author

Keegan Carruthers-Smith

Introduction

You walk past Henry Wensleydale's Cheese Shop and think that a little fermented curd will do the trick in curing your hunger. You enter the shop and ask for some cheese. Wensleydale, the shop owner, says he refuses to help you unless you can solve his problem with cheese lists.

To reduce the costs of printing his cheese lists, he can tell the printer to make a stamp for stamping a word or part of a word. Everyone knows that stamping is cheaper than printing, so if you find the longest prefix shared by at least two cheese types, you can save money by stamping the prefix instead of printing it.

Task

You are given a list of words. You must output the longest prefix shared by at least two words (the prefix may be the whole of one of the words). The longest shared prefix will always exist, and will always be at least one character long. If there is more than one possible output, you must output the prefix that comes first alphabetically.

Example

In the example input there are five prefixes that are shared by at least two words. These are C, CH, CHE, CHA, P. You would output CHA because it is the longest and is alphabetically less than CHE.

Input (lprefix.in)

The first line contains a single integer N. The next N lines each contain one word, W_i. These words contain only uppercase letters, and no two words will be the same.

Sample input

```
7
CHEDDAR
CHESSO
CHAOURCE
PARMESAN
CHAUMES
ROQUEFORT
POSSIA
```

Output (lprefix.out)

The output is a line containing the single prefix L, which is the longest shared prefix. If there is more than one possible L, output the first one when they are arranged alphabetically.

Sample output

```
CHA
```

Constraints

- $2 \leq N \leq 5000$
- $2 \leq \text{length}(W_i) \leq 100$
- $1 \leq \text{length}(L) \leq 99$

Additionally, in 50% of the test-cases:

- $2 \leq N \leq 1000$
- $2 \leq \text{length}(W_i) \leq 50$

Time limit

1 second.

Scoring

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

The Knights Who Say Ni

Author

Harry Wiggins

Introduction

The Knights who say..., who until recently said "Ni!" have tasked Arthur and his knights with cutting down the greatest tree in the forest with...a herring! The king has found the largest tree, but now needs to summon all his knights to assist him in cutting down the tree.

Task

The forest can be represented as a grid of blocks, labeled by their (x, y) coordinates, with the greatest tree in the forest in block $(0, 0)$. The blocks are large enough for any number of knights to share a block, and the forest extends infinitely far in every direction. In order to avoid the many dangers of the forest, the knights move in the same way as knights in chess, as known in Figure 1. King Arthur wants to know how many total moves are required for all the knights to reach the tree.

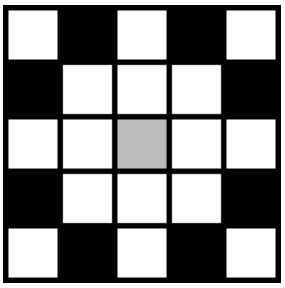

Figure 1: The possible moves a knight can make. The gray block represents the knight's starting position and the black blocks represent the eight positions the knight can move to in one move.

Example

If $N = 2$ and the location of the two knights are $(0, 1)$ and $(2, 1)$, then the knight at $(0, 1)$ needs at least 3 moves whilst the knight at $(2, 1)$ can reach the tree in 1 move. So it will take at least 4 moves to get all the knights to the tree.

Input (knights.in)

The first line of the input contains a single integer, N. The next N lines each contain two space-separated integers, x_i and y_i, being the location of the i^{th} knight.

Sample input

```
4
1 -2
0 2
1 1
-3 1
```

Output (knights.out)

The output is a line containing a single integer, the sum of the minimum number of moves required for each knight to reach the tree.

Sample output

```
7
```

Constraints

- $1 \leq N \leq 10000$
- $-500000 \leq x, y \leq 500000$

Additionally, in 50% of the test-cases:

- $1 \leq N \leq 1000$
- $-500 \leq x, y \leq 500$

Time limit

1 second.

Scoring

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

Trojan Badger

Author

Max Rabkin

Introduction

Having failed to infiltrate the French Castle with the Trojan Rabbit, Sir Bedevere the Wise has decided to give the Trojan Badger a try. The French have become somewhat suspicious, so they aren't about to wheel the Badger into the castle themselves. Fortunately, Sir Bedevere has found a secret entrance to the castle's dungeons. Unfortunately, the Trojan Badger's shape — roughly $1 \times 1 \times 2$ and rectangular — makes it difficult to manoeuvre.

Sir Bedevere's attempts to move the badger across the dungeon are further frustrated by booby traps in the dungeon floor. Fortunately, these are triggered by weight, and if the badger lies across two tiles — even if *both* of them are traps — there is no danger.

Task

Given a map of the dungeon, you must find the minimum number of steps needed to move from the start point to the end point. The badger starts upright, and must end standing upright on the destination square.

Sir Bedevere can move the badger in the following ways, as illustrated in Figure 2:

- If the badger is upright, Sir Bedevere can tip it over.
- If it is lying on the ground, he can stand it up or roll it over.

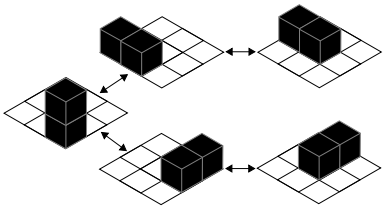

Figure 2: Some of the possible ways Sir Bedevere can move the badger.

Note that when the badger is "standing upright", it might actually be upside-down. Sir Bedevere designed the badger with a flat head, so it can balance equally well on either end, and you needn't bother keeping track of which end is up.

The path you choose may not involve the badger

- leaving the dungeon,
- trying to move through an obstacle, or
- standing upright on a booby trap.

It may involve rolling over the finishing position, as long as it *ends* upright on that position.

Input (trojan.in)

The first line of the input contains two space-separated integers, W and H, representing the size of the dungeon.

The next H lines each consist of W characters. Each character represents a tile in the dungeon, as follows:

Empty floor	`.`
Obstacle	`#`
Trap	`T`
Start	`S`
Finish	`F`

There is exactly one `S` and one `F` in each input file.

Sample input

```
6 3
...#..
...#..
ST..F.
```

Output (trojan.out)

The output consists of a single line, containing the smallest number of steps needed to get from the start position to end position, or -1 if no such path exists.

In the sample case, the badger starts off standing in the south-west corner. He tips it over northward, and rolls it two spaces east. He stands it up southwards, then tips it over and stands it up eastwards. He tips it over again northward, rolls it west, and stands it up onto the finish square, for a total of nine steps.

Sample output

```
9
```

Constraints

- $1 \leq W, H \leq 500$

Additionally, in 50% of the test-cases:

- $1 \leq WH \leq 200$

South African Computer Olympiad
Final Round
Day 1

Time limit

2 seconds.

Scoring

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