UNSW ICPC Workshop T3W1
Discussion Problem Set
Source: South Pacific Divisionals (2018)

Discuss the problems in this document and try to solve them with your group.
Make sure everyone is comfortable with the solution before moving on.
Ask us if you need help, or want to check your solution.

We recommend doing the problems in the given order (roughly difficulty order), but if you don't like a problem feel free to skip it.

If you want to implement the problems afterwards:

1. Register an account at https://prog4fun.csse.canterbury.ac.nz/
2. Go to https://prog4fun.csse.canterbury.ac.nz/mod/quiz/view.php?id=51

Discussion Problem 1

Problem G
God's Number
Time limit: 1 second

The Rubik's Cube is a $3 \times 3 \times 3$ cube puzzle. There are six colours and when solved, the nine squares on each face have the same colour. See Figure G.1.

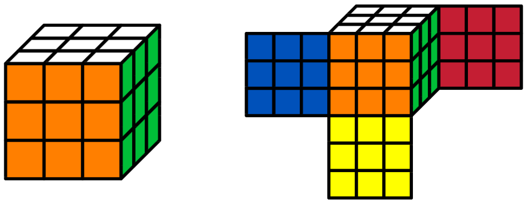
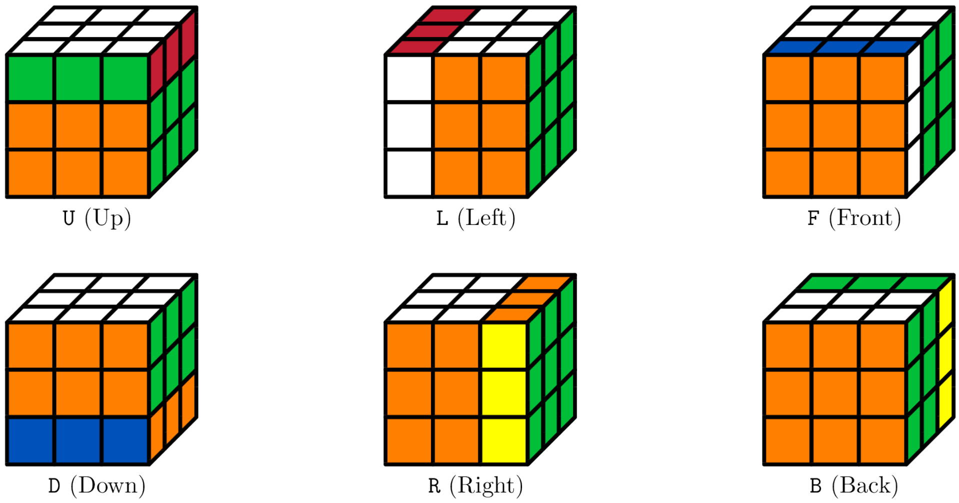


Figure G.1: A solved Rubik's Cube.

Each of the six faces can be rotated either clockwise or counterclockwise, which rotates the colours on that face as well as alters one row on the four neighbouring faces. Here are the six clockwise rotations by 90°:



In 2010, it was proven that any Rubik's Cube configuration can be solved in at most 20 moves (one *move* here is rotating one face by either 90°, 180° or 270° clockwise). In 2014, it was proven that if you were only allowed rotations of 90° or 270° (clockwise), then any Rubik's Cube configuration can be solved in at most 26 moves.

Jacob is attempting to tackle the final case: only allowing rotations of 90° (clockwise). He has been working towards a solution for several days, but cannot seem to even solve the Rubik's Cube! Thankfully, he started with a solved Rubik's Cube and he has kept track of the moves that he has made so far. Given this information, help solve the Rubik's Cube for him using only rotations of 90° (clockwise). You *do not* need to solve the Rubik's Cube in the minimum number of moves.

Input

The input consists of a single line containing a single string, which is the sequence of moves that Jacob has performed on the Rubik's Cube. The string consists of only `B`, `D`, `F`, `L`, `R` and `U`. The Rubik's Cube starts solved, then the sequence of moves are applied, in order. The length of the string is between 1 and 50, inclusive.

Output

Display a sequence of moves that solves the Rubik's Cube. Your solution does not need to be optimal. The sequence of moves must be in the same format as the input. Your sequence can be of any length between 1 and 200, inclusive.

Sample Input 1	Sample Output 1
DUDUDUD	U

Sample Input 2	Sample Output 2
FF	FF

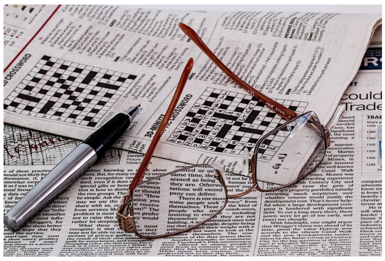
Sample Input 3	Sample Output 3
URRFBRBBRUULBBRUUUDDDRRFRRRLBBUUFF	URRFBRBBRUULBBRUUUDDDRRFRRRLBBUUFF

Sample Input 4	Sample Output 4
FFFF	LLLL

Sample Input 5	Sample Output 5
FB	FFFBBB

Discussion Problem 2

Problem C
Cryptic Clues
Time limit: 3 seconds



Cryptic crosswords contain a variety of clue types. For this problem, we consider *anagram* style clues. For example, `society that reconfigures mac` could be a clue with the solution `acm`.

An anagram is a permutation of a letter sequence. The sequence `mac` can be permuted into `acm`, for example. Letters that occur more than once are considered indistinguishable. A letter sequence is not considered to be an anagram of itself, so `aab` has only two anagrams: `aba` and `baa`.

In this problem, letters in the clue should be permuted to form the solution. These letters must consist of one or more consecutive whole words in the clue, with spaces between words being dropped. For example, in the clue `machine that reuses cut eprom`, the consecutive words `cut eprom` would become `cuteprom` which means a possible solution to the clue is `computer` (`cuteprom` is an anagram of `computer`). Note that `cuteprom` is not a possible solution.

You have a clue. The clue is a sequence of words separated by spaces. You also have a list of candidate solutions. Each of these is a distinct word. Your task is to find the candidate solution corresponding to the clue. This is guaranteed to exist and to be unique.

Input

The input starts with a line containing a single string, which is the clue. The clue contains only lowercase letters with a single space separating words in the clue. A word is a sequence of one or more letters. The total number of letters in the clue is between 2 and 100, inclusive.

The next line contains a single integer W ($1 \leq W \leq 100$), which is the number of candidate solutions.

The next W lines describe the candidate solutions. Each line contains a single word which consists of between 2 and 100 lowercase letters inclusive.

Output

Display the candidate solution matching the clue.

Sample Input 1	Sample Output 1
south pacific icpc 5 icpcsouth pacificicpc icpcpacific southpacific sppc	icpcpacific

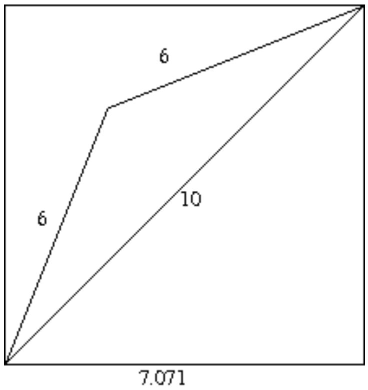
Sample Input 2	Sample Output 2
society that reconfigures mac 1 acm	acm

Sample Input 3	Sample Output 3
hitperson turning off re offer of ref 2 oofer offer	offer

Discussion Problem 3

Problem L
Love Actually
Time limit: 1 second

For Christmas this year, Steve has decided to give his lovely wife her favourite shape, a triangle, framed inside of her second favourite shape, a square. The triangle may be arbitrarily rotated and the entire triangle must fit inside of the square picture frame.



Steve has already purchased the triangle and wants to know the smallest size of a square picture frame that can fit the triangle.

Input

The input consists of a single line containing 3 integers a, b and c ($1 \leq a \leq b \leq c \leq 10\,000$), which are the side lengths of the triangle. It is guaranteed that these values form a valid triangle ($a + b > c$).

Output

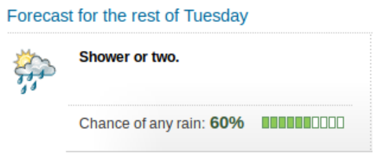
Display the minimum side length of a square that the triangle can fit in, with an absolute or relative error of at most 10^{-9}.

Sample Input 1	Sample Output 1
`6 6 10`	`7.071067811865475`

Sample Input 2	Sample Output 2
`3 4 5`	`3.880570000581328`

Discussion Problem 4

Problem H
Holiday
Time limit: 3 seconds



Your friend, Anna, from America is coming to visit you in Melbourne, Australia. "I hope the weather is nice," she says unaware of what Melbourne has in store for her. She has identified n consecutive days during which she might visit. She does not yet know when she will arrive or how long she will stay for. To prove to her how bad Melbourne's weather is you want to make sure she experiences every possible temperature in the n days.

It has been decided that you will pick the number of days d that Anna will stay for. Anna does not yet know what day she will arrive. However, she will stay for exactly d days when she does arrive (she will never arrive too late to stay for d days). You want to determine the minimum d such that no matter which day Anna arrives, she will experience all of the different temperatures in the n day span. That is, what is the smallest value of d such that no matter which of the first $n - d + 1$ days Anna arrives, she will experience every different temperature at some point during her d day trip?

Input

The first line contains an integer n $(1 \leq n \leq 200\,000)$, which is the number of days.

The second line describes the temperatures. This line contains n integers $t_1, t_2, \ldots, t_n$ $(0 \leq t_i \leq 10^9)$, which are the temperatures on each day in order.

Output

Display the minimum number of days Anna must stay for.

Sample Input 1	Sample Output 1
5 40 10 40 10 40	2

Sample Input 2	Sample Output 2
5 40 40 40 40 40	1

Sample Input 3	Sample Output 3
4 1 2 2 1	3

Sample Input 4	Sample Output 4
6 10 40 20 30 50 4	6

Discussion Problem 5

Problem B
Bombs Ahoy!
Time limit: 6 seconds

Georgie the mouse is sitting in a $N \times M$ grid, worriedly glancing at the sky. This is because he knows that K paint bombs will soon be dropped on the grid from above. Each paint bomb will splatter a rectangular section of the grid with paint.

Georgie does not want his beautiful fur to be painted either by stepping into an already painted square or remaining in a square when a bomb lands. Note that some of the grid squares may already be painted before any bomb has landed.

The K bombs will land in order. Just before each bomb lands, Georgie can move to a neighbouring grid square, or stay in the square he is in. Two squares are neighbouring if they share an edge or a corner. Note that Georgie can move at most once before the next bomb lands. Your task is to help Georgie navigate the grid so that he has no paint on his fur after all the bombs have landed. Georgie stops moving when the last bomb lands.

Input

The input starts with a line containing three integers N ($1 \leq N \leq 100\,000$), which is the number of grid rows, M ($1 \leq M \leq 100\,000$), which is the number of grid columns, and K ($1 \leq K \leq 100\,000$), which is the number of bombs that will be dropped. Furthermore, ($1 \leq N \cdot M \leq 500\,000$).

The next N lines describe the grid. Each line contains a single string of length M consisting only of the characters '`S`', which is the starting position of Georgie, '`X`', which denotes an already painted grid square, and '`.`', which denotes an unpainted grid square.

The next K lines describe the bombs that will be dropped. Each line consists of four integers r_1, c_1, r_2, and c_2 ($1 \leq r_1 \leq r_2 \leq N$ and $1 \leq c_1 \leq c_2 \leq M$). Every grid square within the rectangle defined by these four points will be painted after the bomb is dropped. In other words, a grid square located at (r, c) will be painted if $r_1 \leq r \leq r_2$ and $c_1 \leq c \leq c_2$. Rows are numbered from top to bottom and columns from left to right

Output

If there is no way for Georgie to avoid getting paint on his fur, display `-1`.

If it is possible for Georgie to avoid getting paint on his fur, display the row and column of Georgie's final position. If there are multiple possible final positions, display the position with the smallest row index (breaking ties by selecting the smallest column index).

Sample Input 1

```
5 5 3
..X..
.....
..S..
....X
.....
2 2 4 3
1 4 5 5
5 1 5 2
```

Sample Output 1

```
5 3
```

Sample Input 2	Sample Output 2
5 5 3 ..X.. ....S.. ....X 2 2 4 3 1 3 4 5 5 2 5 5	-1