ICPC Pacific Northwest Region
Division 2

November 9, 2019

Cogswell College
Simon Fraser University
Est. 1888 University of Puget Sound
George Fox University
BYU Hawaii
Whitworth University

Division 2 Problems

Problems

Problem N
Checkerboard
Time limit: 1 second

An R-by-C grid of squares is to be colored in a checkerboard style. The board will be filled with rectangles made up of the grid squares. There should be A rectangles vertically and B rectangles horizontally. All rectangles in row i should be a_i squares high; all rectangles in column j should be b_j squares wide. The top-left rectangle should be black and two adjacent rectangles that share a side should be colored differently.

Print the checkerboard.

Input

The first line will contain four integers, R, C, A, and B with $1 \leq A \leq R \leq 50$ and $1 \leq B \leq C \leq 50$. The next A lines will each contain a single positive integer a_i; the sum of the a_i values will be R. The next B lines will each contain a single positive integer b_i; the sum of the b_i values will be C.

Output

Print the required checkerboard. It should have R lines each with a string of length C containing only the characters '`B`' and '`W`'.

Examples

Sample Input 1

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

Sample Output 1

```
BBBWW
WWWBB
WWWBB
BBBWW
BBBWW
BBBWW
```

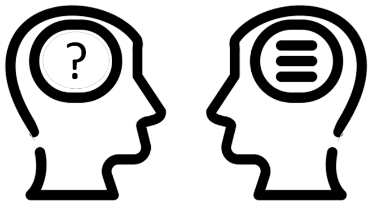
Sample Input 2

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

Sample Output 2

```
BBBW
WWWB
WWWB
WWWB
```

Problem O
Even or Odd
Time limit: 1 second



Your friend has secretly picked N consecutive positive integers between 1 and 100, and wants you to guess if their sum is even or odd.

If the sum must be even, output '`Even`'. If the sum must be odd, output '`Odd`'. If the sum could be even or could be odd, output '`Either`'.

Input

The input is a single integer N with $1 \leq N \leq 10$.

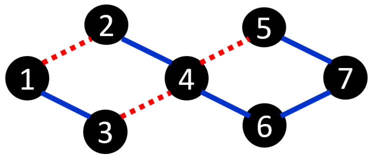
Output

Output a single word. The word should be '`Even`', '`Odd`', or '`Either`', according to the rules given earlier.

Sample Input 1	Sample Output 1
`1`	`Either`

Sample Input 2	Sample Output 2
`2`	`Odd`

Problem P
Coloring Contention
Time limit: 1 second



Alice and Bob are playing a game on a simple connected graph with N nodes and M edges.

Alice colors each edge in the graph red or blue.

A path is a sequence of edges where each pair of consecutive edges have a node in common. If the first edge in the pair is of a different color than the second edge, then that is a “color change.”

After Alice colors the graph, Bob chooses a path that begins at node 1 and ends at node N. He can choose any path on the graph, but he wants to minimize the number of color changes in the path. Alice wants to choose an edge coloring to maximize the number of color changes Bob must make. What is the maximum number of color changes she can force Bob to make, regardless of which path he chooses?

Input

The first line contains two integer values N and M with $2 \leq N \leq 100\,000$ and $1 \leq M \leq 100\,000$. The next M lines contain two integers a_i and b_i indicating an undirected edge between nodes a_i and b_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$).

All edges in the graph are unique.

Output

Output the maximum number of color changes Alice can force Bob to make on his route from node 1 to node N.

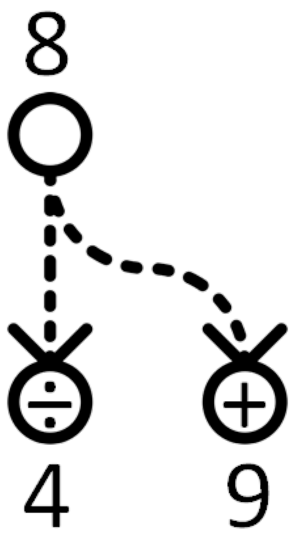
Sample Input 1	Sample Output 1

3 3
1 3
1 2
2 3
	0

Sample Input 2	Sample Output 2

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

Problem Q
Dividing by Two
Time limit: 1 second



You are given two integers, A and B. You want to transform A to B by performing a sequence of operations. You can only perform the following operations:

- Divide A by two, but only if A is even.
- Add one to A.

What is the minimum number of operations you need to transform A into B?

Input

The input will be a single line containing two integers A and B with $1 \leq A, B \leq 10^9$.

Output

On a single line write the minimum number of operations required as an integer.

Examples

Sample Input 1	Sample Output 1
`103 27`	`4`

Sample Input 2	Sample Output 2
`3 8`	`5`

Problem R
Rainbow Strings
Time limit: 1 second

Define a string to be a *rainbow string* if every letter in the string is distinct. An empty string is also considered a *rainbow string*.

Given a string of lowercase letters, compute the number of different subsequences which are *rainbow strings*. Two subsequences are different if an index is included in one subsequence but not the other, even if the resulting strings are identical.

In the first example, there are 8 subsequences. The only subsequences that aren't rainbow strings are `aa` and `aab`. The remaining 6 subsequences are rainbow strings.

Input

The input will consist of a single line with a single string consisting solely of lowercase letters. The length of the string is between 1 and $100\,000$ (inclusive).

Output

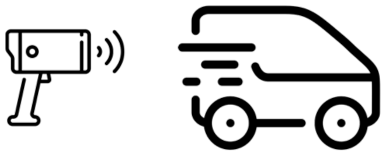
Write on a single line the number of rainbow sequences, modulo the prime $11\,092\,019$.

Examples

Sample Input 1	Sample Output 1
`aab`	`6`

Sample Input 2	Sample Output 2
`icpcprogrammingcontest`	`209952`

Problem S
Speeding
Time limit: 1 second



You'd like to figure out whether a car was speeding while it was driving down a straight road. Unfortunately you don't have any radar guns or related instruments for measuring speed directly; all you have are photographs taken of the car at various checkpoints on the road at various times. Given when and where these photographs were taken, what is the fastest speed that you can prove the car must have been going at some point along the road?

Input

The first line contains an integer N, the number of photographs taken, with $2 \leq N \leq 100$. The following N lines each contain two integers t_i and d_i, with $0 \leq t_i \leq 10\,000$ and $0 \leq d_i \leq 1\,000\,000$. The first photograph is always taken at time 0 with distance 0. Both the times and distances strictly increase. That is, $t_{i+1} > t_i$ and $d_{i+1} > d_i$.

Output

Output the greatest integral speed that you can be certain the car was going at some point.

Examples

Sample Input 1	Sample Output 1

2
0 0
7 42
	6

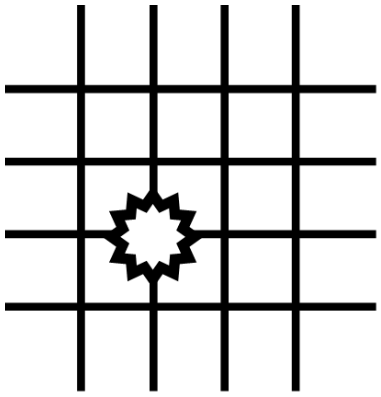
Sample Input 2

```
5
0 0
5 24
10 98
15 222
20 396
```

Sample Output 2

```
34
```

Problem T
Glow, Little Pixel, Glow
Time limit: 2 seconds



An LCD panel is composed of a grid of pixels, spaced 1 alu ("arbitrary length unit") apart both horizontally and vertically. Wires run along each row and each column, intersecting at the pixels. Wires are numbered beginning with 1 and proceeding up to a panel-dependent maximum. The vertical wire numbered 1 lies along the left edge of the panel, and the horizontal wire numbered 1 lies along the bottom edge of the panel.

A pixel will activate, turning dark, when a current is present along both the vertical and horizontal wire passing through that pixel.

For a period of time, we will send pulses of current down selected wires. The current flows down the wires at a speed of one alu per atu ("arbitrary time unit"). The pulses themselves have a length measured in atus. A pixel activates when current is passing through both intersecting wires at the same time. If the leading edge of a pulse on one wire reaches the intersection at the exact same time that the trailing edge of a pulse on the other wire leaves that intersection, the pixel is not activated.

All pulses in vertical wires start from the bottom of the grid. All pulses in horizontal wires start from the left of the grid. At most one pulse will travel along any one wire.

Given the schedule of pulses to be sent through the wires, determine how many pixels will have been activated by the time all pulses have exited the top and right of the grid.

Input

The first line will contain n, the number of current pulses, with $1 \le n \le 200\,000$.

Following this will be n lines, each describing a single pulse. Each such line will contain four elements, separated from one another by a single space:

- A single character that is either ‘h’ or ‘v’, indicating the horizontal/vertical direction of the pulse.
- An integer t, $1 \le t \le 200\,000$, denoting the starting time of the pulse. The starting time is considered to be the moment when the leading edge of a vertical [horizontal] pulse crosses horizontal [vertical] wire #1.
- An integer m, $1 \le m \le 200\,000$, denoting the length of the pulse.
- An integer a, $1 \le a \le 100\,000$, denoting the wire number (horizontal or vertical) along which the pulse will travel.

Output

Print on a single line the number of pixels that will have activated by the time the last pulse of current has left the grid.

Examples

Sample Input 1

```
4
h 1 4 1
v 2 4 2
h 10 2 2
v 11 2 3
```

Sample Output 1

```
2
```

Sample Input 2

```
4
h 1 10 1
h 5 10 2
v 1 10 1
v 5 10 3
```

Sample Output 2

```
4
```

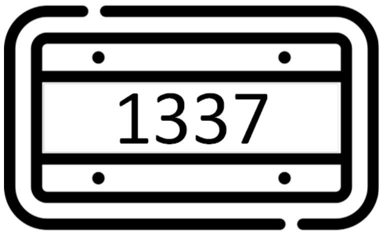
Sample Input 3

```
7
v 1 3 1
v 1 15 2
h 4 5 1
h 5 5 2
h 6 5 3
h 7 5 4
h 8 5 5
```

Sample Output 3

```
5
```

Problem U
Issuing Plates
Time limit: 1 second



You are writing the code to check license plates. These consist of upper letters ‘A’–‘Z’ and numbers ‘0’–‘9’. You want to make sure the codes do not contain any bad words, even considering leetspeak.

Given some input strings, which are valid license plates?

In leetspeak we have the following equivalences: `0=O 1=L 2=Z 3=E 5=S 6=B 7=T 8=B`

Input

The first line will contain the integers N and M with $0 \leq N, M \leq 100$. Following this will be N lines containing bad words; each such word will contain only uppercase alphabetic characters (‘A’–‘Z’) and be at most 25 characters long. Then there will be M lines containing the plates to check; each such plate will consist of only uppercase alphabetic characters and numeric digits (‘0’–‘9’) and be at most 25 characters long.

Output

Your output will be M lines, one per plate, in the same order as the plates are given on the input. If the plate is valid, write out the string ‘`VALID`’; otherwise write out the string ‘`INVALID`’.

Examples

Sample Input 1	Sample Output 1

`7 2`
`AWFUL`
`BAD`
`CRUMMY`
`LOUSY`
`POOR`
`SAD`
`TERRIBLE`
`SO5OD`
`TROUBADOUR`
	`VALID`
	`INVALID`

Problem V
Correcting Keats
Time limit: 1 second

Charles complains, "Keats makes so many spelling errors; it's impossible to fix them all!"

Ada responds, "We have to get the manuscript into the Engine. Maybe we can program the Engine to check each word against a list of English words, and if it's not found in that list, see what sort of small errors might have been made."

Charles asks, "But what is a small error?"

With some discussion they agree on the following list of small errors:

- Adding a letter anywhere in the string.
- Removing a letter from anywhere in the string.
- Changing any letter in the string to any other letter.

Given a specific alphabet and a particular string, find all other strings in that alphabet that can be created by making one of the mistakes in the above list, and list them in alphabetical order.

Note that the input string must not be in the list, and the list must not contain duplicates.

Input

The first line of the input is a sequence of lowercase alphabetic characters, in alphabetical order, with no spaces between them. This is the alphabet to use. The second line contains the input string, which will consist only of letters from the given alphabet, and have length at least 2 and at most 100.

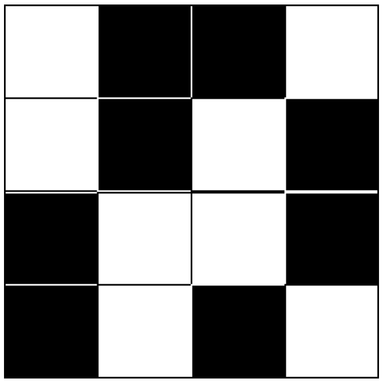
Output

List, in alphabetical order, all strings that can result from making one error in the given word.

Examples

Sample Input 1	Sample Output 1
eg egg	eeg eegg eg ege egeg egge eggg gegg gg ggg

Problem W
Black and White
Time limit: 1 second



You are given an n-by-n grid where each square is colored either black or white. A grid is *correct* if all of the following conditions are satisfied:

- Every row has the same number of black squares as it has white squares.
- Every column has the same number of black squares as it has white squares.
- No row or column has 3 or more consecutive squares of the same color.

Given a grid, determine whether it is *correct*.

Input

The first line contains an integer n ($2 \le n \le 24$; n is even). Each of the next n lines contains a string of length n consisting solely of the characters '`B`' and '`W`', representing the colors of the grid squares.

Output

If the grid is *correct*, print the number 1 on a single line. Otherwise, print the number 0 on a single line.

Sample Input 1

```
4
WBBW
WBWB
BWWB
BWBW
```

Sample Output 1

```
1
```

Sample Input 2

```
4
BWWB
BWBB
WBBW
WBWW
```

Sample Output 2

```
0
```

Sample Input 3

```
6
BWBWWB
WBWBWB
WBBWBW
BBWBWW
BWWBBW
WWBWBB
```

Sample Output 3

```
0
```

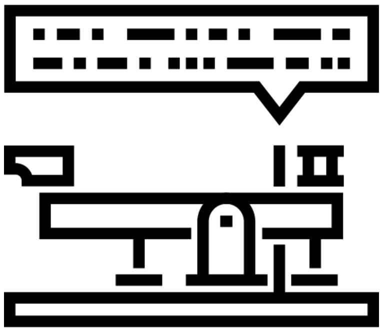
Sample Input 4

```
6
WWBBWB
BBWWBW
WBWBWB
BWBWBW
BWBBWW
WBWWBB
```

Sample Output 4

```
1
```

Problem X
Remorse
Time limit: 1 second



A Morse-like code is an assignment of sequences of dots and dashes to alphabet characters. You are to create a Morse-like code that yields the shortest total length to a given message, and return that total length.

A dot has length 1. A dash has length 3. The gap between dots and dashes within a character has length 1. The gap between characters has length three. Spaces, punctuation, and alphabetic case are ignored. For example, the text

```
The quick brown dog jumps over the lazy fox.
```

is encoded as though it were just

```
THEQUICKBROWNDOGJUMPSOVERTHELAZYFOX
```

For example, with input `ICPC`, the answer is 17: Encode the `C`'s with a single dot, the `I` with a dash, and the `P` with two dots, for a total of '`-   .   . .   .`' which has length 3+3+1+3+1+1+1+3+1 or 17.

Input

The input will be a single line consisting of uppercase or lowercase letters, spaces, commas, periods, exclamation points, and question marks. The line will have a maximum length of 32 000 characters and will contain at least one letter. Everything but the letters should be ignored.

Output

The output will consist of the length of the encoded string when an optimal Morse-like code is used.

Examples

Sample Input 1	Sample Output 1
`ICPC`	`17`

Sample Input 2	Sample Output 2
`A`	`1`

Sample Input 3	Sample Output 3
`The quick brown dog jumps over the lazy fox.`	`335`

Problem Y
Carry Cam Failure
Time limit: 1 second

A Cam from a
Babbage Analytical Engine

"Drat!" cursed Charles. "This stupid carry bar is not working in my Engine! I just tried to calculate the square of a number, but it's wrong; all of the carries are lost."

"Hmm," mused Ada, "arithmetic without carries! I wonder if I can figure out what your original input was, based on the result I see on the Engine."

Carryless addition, denoted by $\oplus$, is the same as normal addition, except any carries are ignored (in base 10). Thus, $37 \oplus 48$ is 75, not 85.

Carryless multiplication, denoted by $\otimes$, is performed using the schoolboy algorithm for multiplication, column by column, but the intermediate additions are calculated using *carryless addition*. More formally, Let $a_m a_{m-1} \ldots a_1 a_0$ be the digits of a, where a_0 is its least significant digit. Similarly define $b_n b_{n-1} \ldots b_1 b_0$ be the digits of b. The digits of $c = a \otimes b$ are given by the following equation:

$$c_k = a_0 b_k \oplus a_1 b_{k-1} \oplus \cdots \oplus a_{k-1} b_1 \oplus a_k b_0,$$

where any a_i or b_j is considered zero if $i > m$ or $j > n$. For example, $9 \otimes 1\,234$ is $9\,876$, $90 \otimes 1\,234$ is $98\,760$, and $99 \otimes 1\,234$ is $97\,536$.

Given N, find the smallest positive integer a such that $a \otimes a = N$.

Input

The input consists of a single line with an integer N, with at most 25 digits and no leading zeros.

Output

Print, on a single line, the least positive number a such that $a \otimes a = N$. If there is no such a, print '`-1`' instead.

Examples

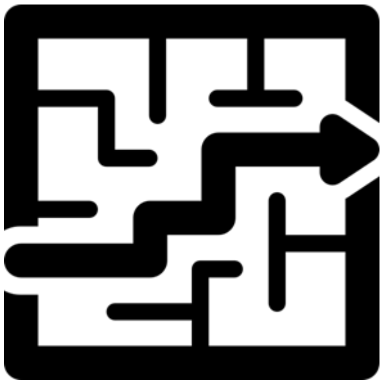
Sample Input 1	Sample Output 1
`6`	`4`

Sample Input 2	Sample Output 2
`149`	`17`

Sample Input 3	Sample Output 3
`123476544`	`11112`

Sample Input 4	Sample Output 4
`15`	`-1`

Problem Z
Maze Connect
Time limit: 5 seconds



Given an orthogonal maze rotated 45 degrees and drawn with forward and backward slash characters (see below), determine the minimum number of walls that need to be removed to ensure it is possible to escape outside of the maze from every square of the (possibly disconnected) maze.

```
/\
\/
```

This maze has only a single square fully enclosed. Removing any wall will connect it to the outside.

```
/\..
\.\.
.\/\
..\/
```

This maze has two enclosed areas. Two walls need to be removed to connect all squares to the outside.

Input

The first line has two numbers, R and C, giving the number of rows and columns in the maze's input description. Following this will be R lines each with C characters, consisting only of the characters '/', '\', and '.'. Both R and C are in the range $1 \ldots 1\,000$.

Define an odd (even) square as one where the sum of the x and y coordinates is odd (even). Either all forward slashes will be in the odd squares and all backslashes in the even squares, or vice versa.

Output

Output on a single line an integer indicating how many walls need to be removed so escape is possible from every square in the maze.

Examples

Sample Input 1
```
2 2
/\
\/
```
Sample Output 1
```
1
```

Sample Input 2
```
4 4
/\..
\.\.
.\/\
..\/
```
Sample Output 2
```
2
```

Sample Input 3
```
2 2
\/
/\
```
Sample Output 3
```
0
```

Sample Input 4
```
8 20
/\/\/\/\/\/\/\/\/\/\
\../\.\/./././\/\/\/
/./\.././\/\.\/\/\/\
\/\/\.\/\/./\/..\../
/\/./\/\/./..\/\/..\
\.\.././\.\/\/./\.\/
/.../\../..\/./.../\
\/\/\/\/\/\/\/\/\/\/
```
Sample Output 4
```
26
```