A. Bayan Bus

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The final round of Bayan Programming Contest will be held in Tehran, and the participants will be carried around with a yellow bus. The bus has 34 passenger seats: 4 seats in the last row and 3 seats in remaining rows.

The event coordinator has a list of k participants who should be picked up at the airport. When a participant gets on the bus, he will sit in the last row with an empty seat. If there is more than one empty seat in that row, he will take the leftmost one.

In order to keep track of the people who are on the bus, the event coordinator needs a figure showing which seats are going to be taken by k participants. Your task is to draw the figure representing occupied seats.

Input

The only line of input contains integer k, $(0 \le k \le 34)$, denoting the number of participants.

Output

Print the figure of a bus with k passengers as described in sample tests. Character '#' denotes an empty seat, while 'O' denotes a taken seat. 'D' is the bus driver and other characters in the output are for the purpose of beautifying the figure. Strictly follow the sample test cases output format. Print exactly six lines. Do not output extra space or other characters.

Sample test(s)

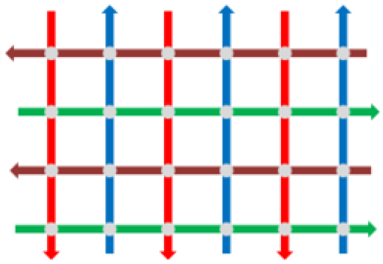
input
```
9
```
output
```
+------------------------+
|O.O.O.#.#.#.#.#.#.#.#.|D|)
|O.O.O.#.#.#.#.#.#.#.#.|.|
|O.......................|
|O.O.#.#.#.#.#.#.#.#.#.|.|)
+------------------------+
```

input
```
20
```
output
```
+------------------------+
|O.O.O.O.O.O.O.#.#.#.#.|D|)
|O.O.O.O.O.O.#.#.#.#.#.|.|
|O.......................|
|O.O.O.O.O.O.#.#.#.#.#.|.|)
+------------------------+
```

B. Strongly Connected City

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Imagine a city with n horizontal streets crossing m vertical streets, forming an $(n - 1) \times (m - 1)$ grid. In order to increase the traffic flow, mayor of the city has decided to make each street one way. This means in each horizontal street, the traffic moves only from west to east or only from east to west. Also, traffic moves only from north to south or only from south to north in each vertical street. It is possible to enter a horizontal street from a vertical street, or vice versa, at their intersection.



The mayor has received some street direction patterns. Your task is to check whether it is possible to reach any junction from any other junction in the proposed street direction pattern.

Input

The first line of input contains two integers n and m, $(2 \le n, m \le 20)$, denoting the number of horizontal streets and the number of vertical streets.

The second line contains a string of length n, made of characters '<' and '>', denoting direction of each horizontal street. If the i-th character is equal to '<', the street is directed from east to west otherwise, the street is directed from west to east. Streets are listed in order from north to south.

The third line contains a string of length m, made of characters '^' and 'v', denoting direction of each vertical street. If the i-th character is equal to '^', the street is directed from south to north, otherwise the street is directed from north to south. Streets are listed in order from west to east.

Output

If the given pattern meets the mayor's criteria, print a single line containing "YES", otherwise print a single line containing "NO".

Sample test(s)

input

```
3 3
><>
v^v
```

output

```
NO
```

input

```
4 6
<><>
v^v^v^
```

output

```
YES
```

Note

The figure above shows street directions in the second sample test case.

C. Kamal-ol-molk's Painting

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Rumors say that one of Kamal-ol-molk's paintings has been altered. A rectangular brush has been moved right and down on the painting.

Consider the painting as a $n \times m$ rectangular grid. At the beginning an $x \times y$ rectangular brush is placed somewhere in the frame, with edges parallel to the frame, $(1 \le x \le n, 1 \le y \le m)$. Then the brush is moved several times. Each time the brush is moved one unit right or down. The brush has been strictly inside the frame during the painting. The brush alters every cell it has covered at some moment.

You have found one of the old Kamal-ol-molk's paintings. You want to know if it's possible that it has been altered in described manner. If yes, you also want to know minimum possible area of the brush.

Input

The first line of input contains two integers n and m, $(1 \le n, m \le 1000)$, denoting the height and width of the painting.

The next n lines contain the painting. Each line has m characters. Character 'X' denotes an altered cell, otherwise it's showed by '.'. There will be at least one altered cell in the painting.

Output

Print the minimum area of the brush in a line, if the painting is possibly altered, otherwise print -1.

Sample test(s)

input

```
4 4
XX..
XX..
XXXX
XXXX
```

output

```
4
```

input

```
4 4
....
.XXX
.XXX
....
```

output

```
2
```

input

```
4 5
XXXX.
XXXX.
.XX..
.XX..
```

output

```
-1
```

D. CGCDSSQ

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given a sequence of integers $a_1, \ldots, a_n$ and q queries $x_1, \ldots, x_q$ on it. For each query x_i you have to count the number of pairs (l, r) such that $1 \le l \le r \le n$ and $gcd(a_l, a_{l+1}, \ldots, a_r) = x_i$.

$\gcd(v_1, v_2, \ldots, v_n)$ is a greatest common divisor of $v_1, v_2, \ldots, v_n$, that is equal to a largest positive integer that divides all v_i.

Input

The first line of the input contains integer n, $(1 \le n \le 10^5)$, denoting the length of the sequence. The next line contains n space separated integers $a_1, \ldots, a_n$, $(1 \le a_i \le 10^9)$.

The third line of the input contains integer q, $(1 \le q \le 3 \times 10^5)$, denoting the number of queries. Then follows q lines, each contain an integer x_i, $(1 \le x_i \le 10^9)$.

Output

For each query print the result in a separate line.

Sample test(s)

input

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

output

```
1
2
2
0
1
```

input

```
7
10 20 3 15 1000 60 16
10
1
2
3
4
5
6
10
20
60
1000
```

output

```
14
0
2
2
2
0
2
2
1
1
```

E. Strongly Connected City 2

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Imagine a city with n junctions and m streets. Junctions are numbered from 1 to n.

In order to increase the traffic flow, mayor of the city has decided to make each street one-way. This means in the street between junctions u and v, the traffic moves only from u to v or only from v to u.

The problem is to direct the traffic flow of streets in a way that maximizes the number of pairs (u, v) where $1 \le u, v \le n$ and it is possible to reach junction v from u by passing the streets in their specified direction. Your task is to find out maximal possible number of such pairs.

Input

The first line of input contains integers n and m, ($1 \le n \le 2000, n-1 \le m \le \frac{n(n-1)}{2}$), denoting the number of junctions and streets of the city.

Each of the following m lines contains two integers u and v, ($u \ne v$), denoting endpoints of a street in the city.

Between every two junctions there will be at most one street. It is guaranteed that before mayor decision (when all streets were two-way) it was possible to reach each junction from any other junction.

Output

Print the maximal number of pairs (u, v) such that that it is possible to reach junction v from u after directing the streets.

Sample test(s)

input
```
5 4
1 2
1 3
1 4
1 5
```
output
```
13
```

input
```
4 5
1 2
2 3
3 4
4 1
1 3
```
output
```
16
```

input
```
2 1
1 2
```
output
```
3
```

input
```
6 7
1 2
2 3
1 3
1 4
4 5
5 6
6 4
```
output
```
27
```

Note

In the first sample, if the mayor makes first and second streets one-way towards the junction 1 and third and fourth streets in opposite direction, there would be 13 pairs of reachable junctions: $\{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (2, 1), (3, 1), (1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)\}$

F. Meta-universe

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Consider infinite grid of unit cells. Some of those cells are *planets*.

Meta-universe $M = \{p_1, p_2, ..., p_k\}$ is a set of planets. Suppose there is an infinite row or column with following two properties: 1) it doesn't contain any planet p_i of meta-universe M on it; 2) there are planets of M located on both sides from this row or column. In this case we can turn the meta-universe M into two non-empty meta-universes M_1 and M_2 containing planets that are located on respective sides of this row or column.

A meta-universe which can't be split using operation above is called a *universe*. We perform such operations until all meta-universes turn to universes.

Given positions of the planets in the original meta-universe, find the number of universes that are result of described process. It can be proved that each universe is uniquely identified not depending from order of splitting.

Input

The first line of input contains an integer n, $(1 \le n \le 10^5)$, denoting the number of planets in the meta-universe.

The next n lines each contain integers x_i and y_i, $(-10^9 \le x_i, y_i \le 10^9)$, denoting the coordinates of the i-th planet. All planets are located in different cells.

Output

Print the number of resulting universes.

Sample test(s)

input

```
5
0 0
0 2
2 0
2 1
2 2
```

output

```
3
```

input

```
8
0 0
1 0
0 2
0 3
3 0
3 1
2 3
3 3
```

output

```
1
```

Note

The following figure describes the first test case: