Codeforces Round #385 (Div. 1)

A. Hongcow Builds A Nation

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

Hongcow is ruler of the world. As ruler of the world, he wants to make it easier for people to travel by road within their own countries.

The world can be modeled as an undirected graph with n nodes and m edges. k of the nodes are home to the governments of the k countries that make up the world.

There is at most one edge connecting any two nodes and no edge connects a node to itself. Furthermore, for any two nodes corresponding to governments, **there is no path between those two nodes**. Any graph that satisfies all of these conditions is *stable*.

Hongcow wants to add as many edges as possible to the graph while keeping it stable. Determine the maximum number of edges Hongcow can add.

Input

The first line of input will contain three integers n, m and k ($1 \le n \le 1\,000$, $0 \le m \le 100\,000$, $1 \le k \le n$) — the number of vertices and edges in the graph, and the number of vertices that are homes of the government.

The next line of input will contain k integers $c_1, c_2, ..., c_k$ ($1 \le c_i \le n$). These integers will be pairwise distinct and denote the nodes that are home to the governments in this world.

The following m lines of input will contain two integers u_i and v_i ($1 \le u_i, v_i \le n$). This denotes an undirected edge between nodes u_i and v_i.

It is guaranteed that the graph described by the input is stable.

Output

Output a single integer, the maximum number of edges Hongcow can add to the graph while keeping it stable.

Examples

input

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

output

```
2
```

input

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

output

```
0
```

Note

For the first sample test, the graph looks like this:

Vertices 1 and 3 are special. The optimal solution is to connect vertex 4 to vertices 1 and 2. This adds a total of 2 edges. We cannot add any more edges, since vertices 1 and 3 cannot have any path between them.
For the second sample test, the graph looks like this:

We cannot add any more edges to this graph. Note that we are not allowed to add self-loops, and the graph must be simple.

B. Hongcow's Game

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is an interactive problem. In the interaction section below you will see the information about flushing the output.

In this problem, you will be playing a game with Hongcow. How lucky of you!

Hongcow has a hidden n by n matrix M. Let $M_{i,j}$ denote the entry i-th row and j-th column of the matrix. The rows and columns are labeled from 1 to n.

The matrix entries are between 0 and 10^9. In addition, $M_{i,i} = 0$ for all valid i. Your task is to find the minimum value along each row, excluding diagonal elements. Formally, for each i, you must find .

To do this, you can ask Hongcow some questions.

A question consists of giving Hongcow a subset of distinct indices $\{w_1, w_2, ..., w_k\}$, with $1 \le k \le n$. Hongcow will respond with n integers. The i-th integer will contain the minimum value of $min_{1 \le j \le k} M_{i, w_j}$.

You may only ask Hongcow at most 20 questions — he thinks you only need that many questions answered.

When you are ready to answer, print out a single integer -1 on its own line, then n integers on the next line. The i-th integer should be the minimum value in the i-th row of the matrix, excluding the i-th element. Do not forget to flush the final answer as well. Printing the answer does not count as asking a question.

You will get `Wrong Answer` verdict if

- Your question or answers are not in the format described in this statement.
- You ask strictly more than 20 questions.
- Your question contains duplicate indices.
- The value of k in your question does not lie in the range from 1 to n, inclusive.
- Your final answer is not correct.

You will get `Idleness Limit Exceeded` if you don't print anything or if you forget to flush the output, including for the final answer (more info about flushing output below).

Input

The first line of input will contain a single integer n ($2 \le n \le 1,000$).

Output

To print the final answer, print out the string `-1` on its own line. Then, the next line should contain n integers. The i-th integer should be the minimum value of the i-th row of the matrix, excluding elements on the diagonal. **Do not forget to flush your answer!**

Interaction

To ask a question, print out a single integer k on its own line, denoting the size of your subset. Then, the next line should contain k integers $w_1, w_2, ...\ w_k$. Note, you must flush your output to get a response.

Hongcow will respond by printing out a line with n integers. The i-th integer in this line represents the minimum value of M_{i, w_j} where j is between 1 and k.

You may only ask a question at most 20 times, otherwise, you will get `Wrong Answer`.

To flush you can use (just after printing an integer and end-of-line):

- `fflush(stdout)` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- `flush(output)` in Pascal;
- See the documentation for other languages.

Hacking To hack someone, use the following format

```
n
M_{1,1} M_{1,2} ... M_{1,n}
M_{2,1} M_{2,2} ... M_{2,n}
...
M_{n,1} M_{n,2} ... M_{n,n}
```

Of course, contestant programs will not be able to see this input.

Examples

input

```
3
0 0 0
2 7 0
0 0 4
3 0 8
0 5 4
```

output

```
3
1 2 3
1
3
2
1 2
1
2
1
1
-1
2 5 4
```

input

```
2
0 0
0 0
```

output

```
1
2
1
1
-1
0 0
```

Note

In the first sample, Hongcow has the hidden matrix

```
[
 [0, 3, 2],
 [5, 0, 7],
 [4, 8 ,0],
]
```

Here is a more readable version demonstrating the interaction. The column on the left represents Hongcow, while the column on the right represents the contestant.

```
3
              3
              1 2 3
0 0 0
              1
              3
2 7 0
              2
              1 2
0 0 4
              1
              2
3 0 8
              1
              1
0 5 4
              -1
              2 5 4
```

For the second sample, it is possible for off-diagonal elements of the matrix to be zero.

C. Hongcow Buys a Deck of Cards

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

One day, Hongcow goes to the store and sees a brand new deck of n special cards. Each individual card is either red or blue. He decides he wants to buy them immediately. To do this, he needs to play a game with the owner of the store.

This game takes some number of turns to complete. On a turn, Hongcow may do one of two things:

- Collect tokens. Hongcow collects 1 red token **and** 1 blue token by choosing this option (thus, 2 tokens in total per one operation).
- Buy a card. Hongcow chooses some card and spends tokens to purchase it as specified below.

The i-th card requires r_i red resources and b_i blue resources. Suppose Hongcow currently has A red cards and B blue cards. Then, the i-th card will require Hongcow to spend $max(r_i - A, 0)$ red tokens, and $max(b_i - B, 0)$ blue tokens. Note, only tokens disappear, but the cards stay with Hongcow forever. Each card can be bought only once.

Given a description of the cards and their costs determine the minimum number of turns Hongcow needs to purchase all cards.

Input

The first line of input will contain a single integer n $(1 \leq n \leq 16)$.

The next n lines of input will contain three tokens c_i, r_i and b_i. c_i will be 'R' or 'B', denoting the color of the card as red or blue. r_i will be an integer denoting the amount of red resources required to obtain the card, and b_i will be an integer denoting the amount of blue resources required to obtain the card $(0 \leq r_i, b_i \leq 10^7)$.

Output

Output a single integer, denoting the minimum number of turns needed to acquire all the cards.

Examples

input

```
3
R 0 1
B 1 0
R 1 1
```

output

```
4
```

input

```
3
R 3 0
R 2 0
R 1 0
```

output

```
6
```

Note

For the first sample, Hongcow's four moves are as follows:

1. Collect tokens
2. Buy card 1
3. Buy card 2
4. Buy card 3

Note, at the fourth step, Hongcow is able to buy card 3 because Hongcow already has one red and one blue card, so we don't need to collect tokens.

For the second sample, one optimal strategy is as follows:

1. Collect tokens
2. Collect tokens
3. Buy card 2
4. Collect tokens
5. Buy card 3
6. Buy card 1

At the fifth step, even though Hongcow has a red token, Hongcow doesn't actually need to spend it, since Hongcow has a red card already.

D. Hongcow Draws a Circle

time limit per test: 6 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Hongcow really likes the color red. Hongcow doesn't like the color blue.

Hongcow is standing in an infinite field where there are n red points and m blue points.

Hongcow wants to draw a circle in the field such that this circle contains at least one red point, and no blue points. Points that line exactly on the boundary of the circle can be counted as either inside or outside.

Compute the radius of the largest circle that satisfies this condition. If this circle can have arbitrarily large size, print -1. Otherwise, your answer will be accepted if it has relative or absolute error at most 10^{-4}.

Input

The first line of the input will contain two integers n, m ($1 \le n, m \le 1,000$).

The next n lines will contain two integers x_i, y_i ($1 \le x_i, y_i \le 10^4$). This denotes the coordinates of a red point.

The next m lines will contain two integers x_i, y_i ($1 \le x_i, y_i \le 10^4$). This denotes the coordinates of a blue point.

No two points will have the same coordinates.

Output

Print -1 if the circle can have arbitrary size. Otherwise, print a floating point number representing the largest radius circle that satisfies the conditions. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-4}.

Namely, let's assume that your answer is a and the answer of the jury is b. The checker program will consider your answer correct if .

Examples

input

```
2 5
2 3
3 4
1 1
1 4
4 2
4 7
2 5
```

output

```
3.5355338827
```

input

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

output

```
1.5811388195
```

input

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

output

```
-1
```

Note

This is a picture of the first sample

This is a picture of the second sample

E. Hongcow Masters the Cyclic Shift

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Hongcow's teacher heard that Hongcow had learned about the cyclic shift, and decided to set the following problem for him.

You are given a list of n strings $s_1, s_2, ..., s_n$ contained in the list A.

A list X of strings is called *stable* if the following condition holds.

First, a *message* is defined as a concatenation of some elements of the list X. You can use an arbitrary element as many times as you want, and you may concatenate these elements in any arbitrary order. Let S_X denote the set of of all messages you can construct from the list. Of course, this set has infinite size if your list is nonempty.

Call a single message *good* if the following conditions hold:

- Suppose the message is the concatenation of k strings $w_1, w_2, ..., w_k$, where each w_i is an element of X.
- Consider the $|w_1| + |w_2| + ... + |w_k|$ cyclic shifts of the string. Let m be the number of these cyclic shifts of the string that are elements of S_X.
- A message is good if and only if m is exactly equal to k.

The list X is called stable if and only if every element of S_X is good.

Let $f(L)$ be 1 if L is a stable list, and 0 otherwise.

Find the sum of $f(L)$ where L is a nonempty **contiguous sublist** of A (there are contiguous sublists in total).

Input

The first line of input will contain a single integer n ($1 \le n \le 30$), denoting the number of strings in the list.

The next n lines will each contain a string s_i ().

Output

Print a single integer, the number of nonempty **contiguous sublists** that are stable.

Examples

input

```
4
a
ab
b
bba
```

output

```
7
```

input

```
5
hh
ee
ll
ll
oo
```

output

```
0
```

input

```
6
aab
ab
bba
b
ab
c
```

output

```
13
```

Note

For the first sample, there are 10 sublists to consider. Sublists ["a", "ab", "b"], ["ab", "b", "bba"], and ["a", "ab", "b", "bba"] are not stable. The other seven sublists are stable.

For example, X = ["a", "ab", "b"] is not stable, since the message "ab" + "ab" = "abab" has four cyclic shifts ["abab", "baba", "abab", "baba"], which are all elements of S_X.

Codeforces (c) Copyright 2010-2017 Mike Mirzayanov
The only programming contests Web 2.0 platform