Codeforces Round #170 (Div. 2)

A. Circle Line

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

The circle line of the Berland subway has n stations. We know the distances between all pairs of neighboring stations:

- d_1 is the distance between the 1-st and the 2-nd station;
- d_2 is the distance between the 2-nd and the 3-rd station;
  ...
- d_{n-1} is the distance between the $n-1$-th and the n-th station;
- d_n is the distance between the n-th and the 1-st station.

The trains go along the circle line in both directions. Find the shortest distance between stations with numbers s and t.

Input

The first line contains integer n $(3 \le n \le 100)$ — the number of stations on the circle line. The second line contains n integers $d_1, d_2, ..., d_n$ $(1 \le d_i \le 100)$ — the distances between pairs of neighboring stations. The third line contains two integers s and t $(1 \le s, t \le n)$ — the numbers of stations, between which you need to find the shortest distance. These numbers can be the same.

The numbers in the lines are separated by single spaces.

Output

Print a single number — the length of the shortest path between stations number s and t.

Sample test(s)

input
4 2 3 4 9 1 3
output
5

input
4 5 8 2 100 4 1
output
15

input
3 1 1 1 3 1
output
1

input
3 31 41 59 1 1
output
0

Note

In the first sample the length of path $1 \to 2 \to 3$ equals 5, the length of path $1 \to 4 \to 3$ equals 13.

In the second sample the length of path $4 \to 1$ is 100, the length of path $4 \to 3 \to 2 \to 1$ is 15.

In the third sample the length of path $3 \to 1$ is 1, the length of path $3 \to 2 \to 1$ is 2.

In the fourth sample the numbers of stations are the same, so the shortest distance equals 0.

B. New Problem

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

Coming up with a new problem isn't as easy as many people think. Sometimes it is hard enough to name it. We'll consider a title *original* if it doesn't occur as a substring in any titles of recent Codeforces problems.

You've got the titles of n last problems — the strings, consisting of lowercase English letters. Your task is to find the shortest original title for the new problem. If there are multiple such titles, choose the lexicographically minimum one. Note, that title of the problem can't be an empty string.

A *substring* $s[l \ldots r]$ $(1 \le l \le r \le |s|)$ of string $s = s_1 s_2 \ldots s_{|s|}$ (where $|s|$ is the length of string s) is string $s_l s_{l+1} \ldots s_r$.

String $x = x_1 x_2 \ldots x_p$ is *lexicographically smaller* than string $y = y_1 y_2 \ldots y_q$, if either $p < q$ and $x_1 = y_1, x_2 = y_2, \ldots, x_p = y_p$, or there exists such number r $(r < p, r < q)$, that $x_1 = y_1, x_2 = y_2, \ldots, x_r = y_r$ and $x_{r+1} < y_{r+1}$. The string characters are compared by their ASCII codes.

Input

The first line contains integer n $(1 \le n \le 30)$ — the number of titles you've got to consider. Then follow n problem titles, one per line. Each title only consists of lowercase English letters (specifically, it doesn't contain any spaces) and has the length from 1 to 20, inclusive.

Output

Print a string, consisting of lowercase English letters — the lexicographically minimum shortest original title.

Sample test(s)

input
5 threehorses goodsubstrings secret primematrix beautifulyear

output
j

input
4 aa bdefghijklmn opqrstuvwxyz c

output
ab

Note

In the first sample the first 9 letters of the English alphabet (`a, b, c, d, e, f, g, h, i`) occur in the problem titles, so the answer is letter `j`.

In the second sample the titles contain 26 English letters, so the shortest original title cannot have length 1. Title `aa` occurs as a substring in the first title.

C. Learning Languages

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

The "BerCorp" company has got n employees. These employees can use m approved official languages for the formal correspondence. The languages are numbered with integers from 1 to m. For each employee we have the list of languages, which he knows. This list could be empty, i. e. an employee may know no official languages. But the employees are willing to learn any number of official languages, as long as the company pays their lessons. A study course in one language for one employee costs 1 berdollar.

Find the minimum sum of money the company needs to spend so as any employee could correspond to any other one (their correspondence can be indirect, i. e. other employees can help out translating).

Input

The first line contains two integers n and m ($2 \le n, m \le 100$) — the number of employees and the number of languages.

Then n lines follow — each employee's language list. At the beginning of the i-th line is integer k_i ($0 \le k_i \le m$) — the number of languages the i-th employee knows. Next, the i-th line contains k_i integers — a_{ij} ($1 \le a_{ij} \le m$) — the identifiers of languages the i-th employee knows. It is guaranteed that all the identifiers in one list are distinct. Note that an employee may know zero languages.

The numbers in the lines are separated by single spaces.

Output

Print a single integer — the minimum amount of money to pay so that in the end every employee could write a letter to every other one (other employees can help out translating).

Sample test(s)

input

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

output

```
0
```

input

```
8 7
0
3 1 2 3
1 1
2 5 4
2 6 7
1 3
2 7 4
1 1
```

output

```
2
```

input

```
2 2
1 2
0
```

output

```
1
```

Note

In the second sample the employee 1 can learn language 2, and employee 8 can learn language 4.

In the third sample employee 2 must learn language 2.

D. Set of Points

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

Convexity of a set of points on the plane is the size of the largest subset of points that form a convex polygon. Your task is to build a set of n points with the convexity of exactly m. Your set of points should not contain three points that lie on a straight line.

Input

The single line contains two integers n and m ($3 \le m \le 100$, $m \le n \le 2m$).

Output

If there is no solution, print "-1". Otherwise, print n pairs of integers — the coordinates of points of any set with the convexity of m. The coordinates shouldn't exceed 10^8 in their absolute value.

Sample test(s)

input
4 3

output
0 0 3 0 0 3 1 1

input
6 3

output
-1

input
6 6

output
10 0 -10 0 10 1 9 1 9 -1 0 -2

input
7 4

output
176166 6377 709276 539564 654734 174109 910147 434207 790497 366519 606663 21061 859328 886001

E. Game

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

Two players play the following game. Initially, the players have a knife and a rectangular sheet of paper, divided into equal square grid cells of unit size. The players make moves in turn, the player who can't make a move loses. In one move, a player can take the knife and cut the paper along any segment of the grid line (not necessarily from border to border). The part of the paper, that touches the knife at least once, is considered cut. There is one limit not to turn the game into an infinite cycle: each move has to cut the paper, that is the knife has to touch the part of the paper that is not cut before.

Obviously, the game ends when the entire sheet is cut into 1×1 blocks. During the game, the pieces of the sheet are not allowed to move. It is also prohibited to cut along the border. The coordinates of the ends of each cut must be integers.

You are given an $n \times m$ piece of paper, somebody has already made k cuts there. Your task is to determine who will win if the players start to play on this sheet. You can consider that both players play optimally well. If the first player wins, you also need to find the winning first move.

Input

The first line contains three integers n, m, k ($1 \le n, m \le 10^9, 0 \le k \le 10^5$) — the sizes of the piece of paper and the number of cuts. Then follow k lines, each containing 4 integers xb_i, yb_i, xe_i, ye_i ($0 \le xb_i, xe_i \le n, 0 \le yb_i, ye_i \le m$) — the coordinates of the ends of the existing cuts.

It is guaranteed that each cut has a non-zero length, is either vertical or horizontal and doesn't go along the sheet border.

The cuts may intersect, overlap and even be the same. That is, it is not guaranteed that the cuts were obtained during any correct game.

Output

If the second player wins, print "SECOND". Otherwise, in the first line print "FIRST", and in the second line print any winning move of the first player (the coordinates of the cut ends, follow input format to print them).

Sample test(s)

input

```
2 1 0
```

output

```
FIRST
1 0 1 1
```

input

```
2 2 4
0 1 2 1
0 1 2 1
1 2 1 0
1 1 1 2
```

output

```
SECOND
```