

Problems

Problem	UHODE	LINJE	SUMA
Source file	uhode.pas uhode.c uhode.cpp	linije.pas linije.c linije.cpp	suma.pas suma.c suma.cpp
Input file	uhode.in	linije.in	suma.in
Output file	uhode.out	linije.out	suma.out
Time limit per test	5 seconds	5 seconds	5 seconds
Number of tests	10	10	10
Points per test	3	5	7
Total points	30	50	70
	150		

UHODE

Luka wasn't able to come to National Competition ten days earlier in order to get some hints by eavesdropping on the The Task Makers' conversations. Fortunately for him, he knows that The Task Makers take daily walks through the town of Cres, during which they quietly discuss about the tasks. So, he decided to put his spy network (consisting of his relatives and good friends) in motion.

Curiously, the city of Cres is built in such a fasion, that it can be represented as a planar coordinate grid. The Task Makers start their walks at point **(0,0)**, and in **every turn they move** on the coordinate plain in **one of the four** directions: right(they increase their x coordinate by one), up(they increase their y coordinate by one), left or down. Luka's spies are positioned on various (fixed) coordinates on the coordinate plain, and they can overhear bits of information **if and only if** they are located on the **same** position as the task masters or one of the other **eight adjacent** positions. To more easily keep track of his spies and to confuse his opponents Luka decided to mark all his spies with an unique number, in such a way that the first spy gets the number 1, the second the number 2, etc.

Your task is to write a program that will find out **which spies** have heard the conversation and **output their numbers**.

Input data

In the first line of the input file is an integer N , $1 \leq N \leq 1000$, the number of spies. In every of the following N lines are two integers, X and Y separated by exactly one space, $-10000 \leq X, Y \leq 10000$, the positions of the spies.

In the next line is an integer K , $1 \leq K \leq 100,000$. The following line holds K charachters that show us the route that The Task Makers were taking that day. Those charachters will be either **'T'** - east (or right), **'S'** - north (or up), **'Z'** - west (or left) or **'J'**-south (or down).

Output data

In the output file you have to write the numbers of the spies that have successfully overheard The Task Masters. The numbers have be outputted in a **strictly ascending** order, and each one of them has to be in a **separate** line.

If no spies were able to hear The Task Masters, in the first and only line of the output file you have to write **'-1'**.

Examples

uhode.in

```
4
-1 -3
3 -1
-3 3
3 3
11
SZJJISSZZ
```

uhode.out

```
-1
```

uhode.in

```
3
2 0
-1 3
-2 -1
10
SISSZZZJII
```

uhode.out

```
1
2
```

uhode.in

```
5
1 2
1 3
1 4
1 5
0 5
5
SSZSS
```

uhode.out

```
1
2
5
```

LINJE

The tram network in my town is made of tram stations and lines that connect them.

One tram line consists of a sequence of distinct stations and there is always a direct line between two consecutive stations on a line. Every two stations are connected with a line in a **unique** way. Also, there is always a way to get from one train station to another by using the city tram.

In an effort to help their citizens our diligent town officials have decided to paint every tram with a colour (marked by numbers from 1 to B) so all the trams on the same line have the **same** colour, but all the trams that pass through the **same** station must be **differently** coloured.

Since our town's budget is very small, we need to use as little of different colours as possible.

Please, please help us! We need a program that will determine some way to paint our trams, so that the number different colours is **minimal**.

Input data

In the first line of the input are two integers, N and M separated by a single space, $1 \leq N \leq 1000$, $1 \leq M \leq 20000$, the number of stations and the number of tram lines.

The next M lines describe the tram lines.

Each description begins with an integer R, the number of the tram stations through which the line passes (including the start and stop stations, which are mutually different). All the numbers in the descriptions are separated by a single space.

Note: the size of the input file will always be smaller than 2 MB.

Output data

In the first line of the output file you have to write the minimal number of the colours used to that will be used paint our town.

In the second line you have to write M numbers, separated by a single space. These numbers should present one of the possible ways to paint the trams, each number for one station, the first number is the colour of the first station etc.

Examples

linije.in

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

linije.out

```
2
1 1 2
```

linije.in

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

linije.out

```
3
1 2 3 2
```

linije.in

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

linije.out

```
2
1 2 1 2
```

SUMA

We can represent a 2D vector as a pair (X,Y).

The sum of two or more vectors is a vector whose coordinates are the sums of the corresponding coordinates of all the vectors in the sum.

e.g. $(1,2)+(3,4)+(5,6) = (1+3+5,2+4+6) = (9,12)$

Weight of a vector (x,y) is defined as $x*x+y*y$.

You are given N vectors on a plain.

Your task is to write a program that will determine a subset of those vectors so the weight of the sum of all vectors in that subset is **maximal**.

Note: Use 64-bit integers (int64 in pascal or long long in c)

Input data

In the first line of the input file is an integer N, $1 \leq N \leq 30,000$, the number of vectors.

The following N lines contain descriptions for each of the vectors.

A description is made of two integers X and Y, separated by a single blank, $-30,000 \leq X,Y \leq 30,000$.

None of the given vectors will be (0,0).

Output data

In the first and only line of the output file you have to write the weight of the maximum sum.

Examples

`suma.in`

```
5
5 -8
-4 2
4 -2
2 1
-6 4
```

`suma.out`

```
202
```

`suma.in`

```
4
1 4
-1 -1
1 -1
-1 4
```

`suma.out`

```
64
```

`suma.in`

```
9
0 1
6 8
0 -1
0 6
-1 1
-1 2
5 -4
1 0
6 -5
```

`suma.out`

```
360
```