A. Amusing Joke

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

So, the New Year holidays are over. Santa Claus and his colleagues can take a rest and have guests at last. When two "New Year and Christmas Men" meet, thear assistants cut out of cardboard the letters from the guest's name and the host's name in honor of this event. Then the hung the letters above the main entrance. One night, when everyone went to bed, someone took all the letters of our characters' names. Then he may have shuffled the letters and put them in one pile in front of the door.

The next morning it was impossible to find the culprit who had made the disorder. But everybody wondered whether it is possible to restore the names of the host and his guests from the letters lying at the door? That is, we need to verify that there are no extra letters, and that nobody will need to cut more letters.

Help the "New Year and Christmas Men" and their friends to cope with this problem. You are given both inscriptions that hung over the front door the previous night, and a pile of letters that were found at the front door next morning.

Input

The input file consists of three lines: the first line contains the guest's name, the second line contains the name of the residence host and the third line contains letters in a pile that were found at the door in the morning. All lines are not empty and contain only uppercase Latin letters. The length of each line does not exceed 100.

Output

Print "YES" without the quotes, if the letters in the pile could be permuted to make the names of the "New Year and Christmas Men". Otherwise, print "NO" without the quotes.

Sample test(s)

input

```
SANTACLAUS
DEDMOROZ
SANTAMOROZDEDCLAUS
```

output

```
YES
```

input

```
PAPAINOEL
JOULUPUKKI
JOULNAPAOILELUPUKKI
```

output

```
NO
```

input

```
BABBONATALE
FATHERCHRISTMAS
BABCHRISTMASBONATALLEFATHER
```

output

```
NO
```

Note

In the first sample the letters written in the last line can be used to write the names and there won't be any extra letters left.

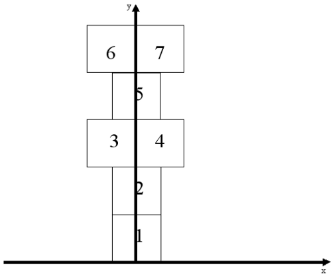
In the second sample letter "P" is missing from the pile and there's an extra letter "L".

In the third sample there's an extra letter "L".

B. Hopscotch

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

So nearly half of the winter is over and Maria is dreaming about summer. She's fed up with skates and sleds, she was dreaming about Hopscotch all night long. It's a very popular children's game. The game field, the court, looks as is shown in the figure (all blocks are square and are numbered from bottom to top, blocks in the same row are numbered from left to right). Let us describe the hopscotch with numbers that denote the number of squares in the row, staring from the lowest one: 1-1-2-1-2-1-2-(1-2)..., where then the period is repeated (1-2).



The coordinate system is defined as shown in the figure. Side of all the squares are equal and have length a.

Maria is a very smart and clever girl, and she is concerned with quite serious issues: if she throws a stone into a point with coordinates (x, y), then will she hit some square? If the answer is positive, you are also required to determine the number of the square.

It is believed that the stone has fallen into the square if it is located **strictly** inside it. In other words a stone that has fallen on the square border is not considered a to hit a square.

Input

The only input line contains three integers: a, x, y, where a $(1 \le a \le 100)$ is the side of the square, x and y $(-10^6 \le x \le 10^6, 0 \le y \le 10^6)$ are coordinates of the stone.

Output

Print the number of the square, inside which the stone fell. If the stone is on a border of some stone or outside the court, print "`-1`" without the quotes.

Sample test(s)

input
```
1 0 0
```
output
```
-1
```

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

input
```
3 0 10
```
output
```
5
```

input
```
3 0 7
```
output
```
-1
```

input
```
3 4 0
```
output
```
-1
```

C. Queue

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

In the Main Berland Bank n people stand in a queue at the cashier, everyone knows his/her height h_i, and the heights of the other people in the queue. Each of them keeps in mind number a_i — how many people who are taller than him/her and stand in queue in front of him.

After a while the cashier has a lunch break and the people in the queue seat on the chairs in the waiting room in a random order.

When the lunch break was over, it turned out that nobody can remember the exact order of the people in the queue, but everyone remembers his number a_i.

Your task is to restore the order in which the people stood in the queue if it is possible. There may be several acceptable orders, but you need to find any of them. Also, you need to print a possible set of numbers h_i — the heights of people in the queue, so that the numbers a_i are correct.

Input

The first input line contains integer n — the number of people in the queue ($1 \le n \le 3000$). Then n lines contain descriptions of the people as "$name_i$ a_i" (one description on one line), where $name_i$ is a non-empty string consisting of lowercase Latin letters whose length does not exceed 10 characters (the i-th person's name), a_i is an integer ($0 \le a_i \le n - 1$), that represents the number of people who are higher and stand in the queue in front of person i. It is guaranteed that all names are different.

Output

If there's no acceptable order of the people in the queue, print the single line containing "`-1`" without the quotes. Otherwise, print in n lines the people as "$name_i$ h_i", where h_i is the integer from 1 to 10^9 (inclusive), the possible height of a man whose name is $name_i$. Print the people in the order in which they stand in the queue, starting from the head of the queue and moving to its tail. Numbers h_i are not necessarily unique.

Sample test(s)

input

```
4
a 0
b 2
c 0
d 0
```

output

```
a 150
c 170
d 180
b 160
```

input

```
4
vasya 0
petya 1
manya 3
dunay 3
```

output

```
-1
```

D. Take-off Ramps

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya participates in a ski race along the X axis. The start is at point 0, and the finish is at L, that is, at a distance L meters from the start in the positive direction of the axis. Vasya has been training so hard that he can run one meter in exactly one second.

Besides, there are n take-off ramps on the track, each ramp is characterized by four numbers:

- x_i represents the ramp's coordinate
- d_i represents from how many meters Vasya will land if he goes down this ramp
- t_i represents the flight time in seconds
- p_i is the number, indicating for how many meters Vasya should gather speed to get ready and fly off the ramp. As Vasya gathers speed, he should ski on the snow (that is, he should not be flying), but his speed still equals one meter per second.

Vasya is allowed to move in **any direction** on the X axis, but he is prohibited to cross the start line, that is go to the negative semiaxis. Vasya himself chooses which take-off ramps he will use and in what order, that is, he is not obliged to take off from all the ramps he encounters. Specifically, Vasya can skip the ramp. It is guaranteed that $x_i + d_i \le L$, that is, Vasya cannot cross the finish line in flight.

Vasya can jump from the ramp only in the positive direction of X axis. More formally, when using the i-th ramp, Vasya starts gathering speed at point x_i - p_i, jumps at point x_i, and lands at point $x_i + d_i$. He cannot use the ramp in opposite direction.

Your task is to find the minimum time that Vasya will spend to cover the distance.

Input

The first line contains two integers n and L ($0 \le n \le 10^5$, $1 \le L \le 10^9$). Then n lines contain the descriptions of the ramps, each description is on a single line. Each description is a group of four non-negative integers x_i, d_i, t_i, p_i ($0 \le x_i \le L$, $1 \le d_i, t_i, p_i \le 10^9$, $x_i + d_i \le L$).

Output

Print in the first line the minimum time in seconds Vasya needs to complete the track. Print in the second line k — the number of take-off ramps that Vasya needs to use, and print on the third line of output k numbers the number the take-off ramps Vasya used in the order in which he used them. Print each number exactly once, separate the numbers with a space. The ramps are numbered starting from 1 in the order in which they are given in the input.

Sample test(s)

input

```
2 20
5 10 5 5
4 16 1 7
```

output

```
15
1
1
```

input

```
2 20
9 8 12 6
15 5 1 1
```

output

```
16
1
2
```

Note

In the first sample, Vasya cannot use ramp 2, because then he will need to gather speed starting from point `-3`, which is not permitted by the statement. The optimal option is using ramp 1, the resulting time is: moving to the point of gathering speed + gathering speed until reaching the takeoff ramp + flight time + moving to the finish line $= 0 + 5 + 5 + 5 = 15$.

In the second sample using ramp 1 is not optimal for Vasya as $t_1 > d_1$. The optimal option is using ramp 2, the resulting time is: moving to the point of gathering speed + gathering speed until reaching the takeoff ramp + flight time + moving to the finish line $= 14 + 1 + 1 + 0 = 16$.

E. Clearing Up

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

After Santa Claus and his assistant Elf delivered all the presents and made all the wishes come true, they returned to the North Pole and found out that it is all covered with snow. Both of them were quite tired and they decided only to remove the snow from the roads connecting huts. The North Pole has n huts connected with m roads. One can go along the roads in both directions.

The Elf offered to split: Santa Claus will clear up the wide roads and the Elf will tread out the narrow roads. For each road they decided who will clear it: Santa Claus or the Elf. To minimize the efforts they decided to clear the road so as to fulfill both those conditions:

- between any two huts should exist **exactly one simple path** along the cleared roads;
- Santa Claus and the Elf should clear the same number of roads.

At this point Santa Claus and his assistant Elf wondered which roads should they clear up?

Input

The first input line contains two positive integers n and m $(1 \le n \le 10^3,\ 1 \le m \le 10^5)$ — the number of huts and the number of roads. Then follow m lines, each of them contains a road description: the numbers of huts it connects — x and y $(1 \le x, y \le n)$ and the person responsible for clearing out this road ("`S`" — for the Elf or "`M`" for Santa Claus). It is possible to go on each road in both directions. Note that there can be more than one road between two huts and a road can begin and end in the same hut.

Output

Print "`-1`" without the quotes if it is impossible to choose the roads that will be cleared by the given rule. Otherwise print in the first line how many roads should be cleared and in the second line print the numbers of those roads (the roads are numbered from 1 in the order of occurrence in the input). It is allowed to print the numbers of the roads in any order. Each number should be printed exactly once. As you print the numbers, separate them with spaces.

Sample test(s)

input

```
1 2
1 1 S
1 1 M
```

output

```
0
```

input

```
3 3
1 2 S
1 3 M
2 3 S
```

output

```
2
2 1
```

input

```
5 6
1 1 S
1 2 M
1 3 S
1 4 M
1 5 M
2 2 S
```

output

```
-1
```

Note

A path is called simple if all huts on it are pairwise different.