Codeforces β
Sponsored by Telegram

Codeforces Round #357 (Div. 2)

A. A Good Contest

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Codeforces user' handle color depends on his rating — it is red if his rating is greater or equal to 2400; it is orange if his rating is less than 2400 but greater or equal to 2200, etc. Each time participant takes part in a rated contest, his rating is changed depending on his performance.

Anton wants the color of his handle to become red. He considers his performance in the rated contest to be *good* if he outscored some participant, whose handle was colored red before the contest and his rating has increased after it.

Anton has written a program that analyses contest results and determines whether he performed good or not. Are you able to do the same?

Input

The first line of the input contains a single integer n $(1 \le n \le 100)$ — the number of participants Anton has outscored in this contest .

The next n lines describe participants results: the i-th of them consists of a participant handle $name_i$ and two integers $before_i$ and $after_i$ $(-4000 \le before_i, after_i \le 4000)$ — participant's rating before and after the contest, respectively. Each handle is a non-empty string, consisting of no more than 10 characters, which might be lowercase and uppercase English letters, digits, characters «_» and «-» characters.

It is guaranteed that all handles are distinct.

Output

Print «YES» (quotes for clarity), if Anton has performed good in the contest and «NO» (quotes for clarity) otherwise.

Examples

input

```
3
Burunduk1 2526 2537
BudAlNik 2084 2214
subscriber 2833 2749
```

output

```
YES
```

input

```
3
Applejack 2400 2400
Fluttershy 2390 2431
Pinkie_Pie -2500 -2450
```

output

```
NO
```

Note

In the first sample, Anton has outscored user with handle Burunduk1, whose handle was colored red before the contest and his rating has increased after the contest.

In the second sample, Applejack's rating has not increased after the contest, while both Fluttershy's and Pinkie_Pie's handles were not colored red before the contest.

B. Economy Game

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Kolya is developing an economy simulator game. His most favourite part of the development process is in-game testing. Once he was entertained by the testing so much, that he found out his game-coin score become equal to 0.

Kolya remembers that at the beginning of the game his game-coin score was equal to n and that he have bought only some houses (for $1\,234\,567$ game-coins each), cars (for $123\,456$ game-coins each) and computers (for $1\,234$ game-coins each).

Kolya is now interested, whether he could have spent all of his initial n game-coins buying only houses, cars and computers or there is a bug in the game. Formally, is there a triple of non-negative integers a, b and c such that $a \times 1\,234\,567 + b \times 123\,456 + c \times 1\,234 = n$?

Please help Kolya answer this question.

Input

The first line of the input contains a single integer n ($1 \le n \le 10^9$) — Kolya's initial game-coin score.

Output

Print "YES" (without quotes) if it's possible that Kolya spent all of his initial n coins buying only houses, cars and computers. Otherwise print "NO" (without quotes).

Examples

input

```
1359257
```

output

```
YES
```

input

```
17851817
```

output

```
NO
```

Note

In the first sample, one of the possible solutions is to buy one house, one car and one computer, spending $1\,234\,567 + 123\,456 + 1234 = 1\,359\,257$ game-coins in total.

C. Heap Operations

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Petya has recently learned data structure named "Binary heap".

The heap he is now operating with allows the following operations:

- put the given number into the heap;
- get the value of the minimum element in the heap;
- extract the minimum element from the heap;

Thus, at any moment of time the heap contains several integers (possibly none), some of them might be equal.

In order to better learn this data structure Petya took an empty heap and applied some operations above to it. Also, he carefully wrote down all the operations and their results to his event log, following the format:

- `insert` x — put the element with value x in the heap;
- `getMin` x — the value of the minimum element contained in the heap was equal to x;
- `removeMin` — the minimum element was extracted from the heap (only one instance, if there were many).

All the operations were correct, i.e. there was at least one element in the heap each time `getMin` or `removeMin` operations were applied.

While Petya was away for a lunch, his little brother Vova came to the room, took away some of the pages from Petya's log and used them to make paper boats.

Now Vova is worried, if he made Petya's sequence of operations inconsistent. For example, if one apply operations one-by-one in the order they are written in the event log, results of `getMin` operations might differ from the results recorded by Petya, and some of `getMin` or `removeMin` operations may be incorrect, as the heap is empty at the moment they are applied.

Now Vova wants to add some new operation records to the event log in order to make the resulting sequence of operations correct. That is, the result of each `getMin` operation is equal to the result in the record, and the heap is non-empty when `getMin` ad `removeMin` are applied. Vova wants to complete this as fast as possible, as the Petya may get back at any moment. He asks you to add the least possible number of operation records to the current log. Note that arbitrary number of operations may be added at the beginning, between any two other operations, or at the end of the log.

Input

The first line of the input contains the only integer n ($1 \leq n \leq 100\,000$) — the number of the records left in Petya's journal.

Each of the following n lines describe the records in the current log in the order they are applied. Format described in the statement is used. All numbers in the input are integers not exceeding 10^9 by their absolute value.

Output

The first line of the output should contain a single integer m — the minimum possible number of records in the modified sequence of operations.

Next m lines should contain the corrected sequence of records following the format of the input (described in the statement), one per line and in the order they are applied. All the numbers in the output should be integers not exceeding 10^9 by their absolute value.

Note that the input sequence of operations must be the **subsequence** of the output sequence.

It's guaranteed that there exists the correct answer consisting of no more than $1\,000\,000$ operations.

Examples

input

```
2
insert 3
getMin 4
```

output

```
4
insert 3
removeMin
insert 4
getMin 4
```

input

```
4
insert 1
insert 1
removeMin
getMin 2
```

output

```
6
insert 1
```

```
insert 1
removeMin
removeMin
insert 2
getMin 2
```

Note

In the first sample, after number 3 is inserted into the heap, the minimum number is 3. To make the result of the first `getMin` equal to 4 one should firstly remove number 3 from the heap and then add number 4 into the heap.

In the second sample case number 1 is inserted two times, so should be similarly removed twice.

D. Gifts by the List

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Sasha lives in a big happy family. At the Man's Day all the men of the family gather to celebrate it following their own traditions. There are n men in Sasha's family, so let's number them with integers from 1 to n.

Each man has at most one father but may have arbitrary number of sons.

Man number A is considered to be the *ancestor* of the man number B if at least one of the following conditions is satisfied:

- $A = B$;
- the man number A is the father of the man number B;
- there is a man number C, such that the man number A is his ancestor and the man number C is the father of the man number B.

Of course, if the man number A is an ancestor of the man number B and $A \neq B$, then the man number B is not an ancestor of the man number A.

The tradition of the Sasha's family is to give gifts at the Man's Day. Because giving gifts in a normal way is boring, each year the following happens.

1. A list of candidates is prepared, containing some (possibly all) of the n men in some order.
2. Each of the n men decides to give a gift.
3. In order to choose a person to give a gift to, man A looks through the list and picks the first man B in the list, such that B is an ancestor of A and gives him a gift. Note that according to definition it may happen that a person gives a gift to himself.
4. If there is no ancestor of a person in the list, he becomes sad and leaves the celebration without giving a gift to anyone.

This year you have decided to help in organizing celebration and asked each of the n men, who do they want to give presents to (this person is chosen only among ancestors). Are you able to make a list of candidates, such that all the wishes will be satisfied if they give gifts according to the process described above?

Input

In the first line of the input two integers n and m ($0 \leq m < n \leq 100\,000$) are given — the number of the men in the Sasha's family and the number of family relations in it respectively.

The next m lines describe family relations: the $(i+1)^{th}$ line consists of pair of integers p_i and q_i ($1 \leq p_i, q_i \leq n$, $p_i \neq q_i$) meaning that the man numbered p_i is the father of the man numbered q_i. It is guaranteed that every pair of numbers appears at most once, that among every pair of two different men at least one of them is not an ancestor of another and that every man has at most one father.

The next line contains n integers $a_1, a_2, ..., a_n$ ($1 \leq a_i \leq n$), i^{th} of which means that the man numbered i wants to give a gift to the man numbered a_i. It is guaranteed that for every $1 \leq i \leq n$ the man numbered a_i is an ancestor of the man numbered i.

Output

Print an integer k $(1 \leq k \leq n)$ — the number of the men in the list of candidates, in the first line.

Print then k pairwise different positive integers not exceeding n — the numbers of the men in the list in an order satisfying every of the men's wishes, one per line.

If there are more than one appropriate lists, print any of them. If there is no appropriate list print -1 in the only line.

Examples

input

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

output

```
-1
```

input

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

output

```
3
2
1
3
```

Note

The first sample explanation:

- if there would be no 1 in the list then the first and the third man's wishes would not be satisfied $(a_1 = a_3 = 1)$;
- if there would be no 2 in the list then the second man wish would not be satisfied $(a_2 = 2)$;
- if 1 would stay before 2 in the answer then the second man would have to give his gift to the first man, but he wants to give it to himself $(a_2 = 2)$.
- if, at the other hand, the man numbered 2 would stay before the man numbered 1, then the third man would have to give his gift to the second man, but not to the first $(a_3 = 1)$.

E. Runaway to a Shadow

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dima is living in a dormitory, as well as some cockroaches.

At the moment 0 Dima saw a cockroach running on a table and decided to kill it. Dima needs exactly T seconds for aiming, and after that he will precisely strike the cockroach and finish it.

To survive the cockroach has to run into a shadow, cast by round plates standing on the table, in T seconds. Shadow casted by any of the plates has the shape of a circle. Shadow circles may intersect, nest or overlap arbitrarily.

The cockroach uses the following strategy: first he equiprobably picks a direction to run towards and then runs towards it with the constant speed v. If at some moment $t \le T$ it reaches any shadow circle, it immediately stops in the shadow and thus will stay alive. Otherwise the cockroach is killed by the Dima's precise strike. Consider that the Dima's precise strike is instant.

Determine the probability of that the cockroach will stay alive.

Input

In the first line of the input the four integers x_0, y_0, v, T ($|x_0|, |y_0| \le 10^9$, $0 \le v, T \le 10^9$) are given — the cockroach initial position on the table in the Cartesian system at the moment 0, the cockroach's constant speed and the time in seconds Dima needs for aiming respectively.

In the next line the only number n ($1 \le n \le 100\ 000$) is given — the number of shadow circles casted by plates.

In the next n lines shadow circle description is given: the i^{th} of them consists of three integers x_i, y_i, r_i ($|x_i|, |y_i| \le 10^9$, $0 \le r \le 10^9$) — the i^{th} shadow circle on-table position in the Cartesian system and its radius respectively.

Consider that the table is big enough for the cockroach not to run to the table edges and avoid Dima's precise strike.

Output

Print the only real number p — the probability of that the cockroach will stay alive.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-4}.

Examples

input

```
0 0 1 1
3
1 1 1
-1 -1 1
-2 2 1
```

output

```
0.50000000000
```

input

```
0 0 1 0
1
1 0 1
```

output

```
1.00000000000
```

Note

The picture for the first sample is given below.

Red color stands for points which being chosen as the cockroach's running direction will cause him being killed, green color for those standing for survival directions. Please note that despite containing a circle centered in $(-2, 2)$ a part of zone is colored red because the cockroach is not able to reach it in one second.