A. Optimal Currency Exchange

time limit per test: 1.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Andrew was very excited to participate in Olympiad of Metropolises. Days flew by quickly, and Andrew is already at the airport, ready to go home. He has n rubles left, and would like to exchange them to euro and dollar bills. Andrew can mix dollar bills and euro bills in whatever way he wants. The price of one dollar is d rubles, and one euro costs e rubles.

Recall that there exist the following dollar bills: 1, 2, 5, 10, 20, 50, 100, and the following euro bills — 5, 10, 20, 50, 100, 200 (note that, in this problem we do **not** consider the 500 euro bill, it is hard to find such bills in the currency exchange points). Andrew can buy any combination of bills, and his goal is to minimize the total number of rubles he will have after the exchange.

Help him — write a program that given integers n, e and d, finds the minimum number of rubles Andrew can get after buying dollar and euro bills.

Input

The first line of the input contains one integer n ($1 \leq n \leq 10^8$) — the initial sum in rubles Andrew has.

The second line of the input contains one integer d ($30 \leq d \leq 100$) — the price of one dollar in rubles.

The third line of the input contains integer e ($30 \leq e \leq 100$) — the price of one euro in rubles.

Output

Output one integer — the minimum number of rubles Andrew can have after buying dollar and euro bills optimally.

Examples

input

```
100
60
70
```

output

```
40
```

input

```
410
55
70
```

output

```
5
```

input

```
600
60
70
```

output

```
0
```

Note

In the first example, we can buy just 1 dollar because there is no 1 euro bill.

In the second example, optimal exchange is to buy 5 euro and 1 dollar.

In the third example, optimal exchange is to buy 10 dollars in one bill.

B. Badges

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

There are b boys and g girls participating in Olympiad of Metropolises. There will be a board games tournament in the evening and n participants have accepted the invitation. The organizers do not know how many boys and girls are among them.

Organizers are preparing red badges for girls and blue ones for boys.

Vasya prepared $n+1$ decks of badges. The i-th (where i is from 0 to n, inclusive) deck contains i blue badges and $n-i$ red ones. The total number of badges in any deck is exactly n.

Determine the **minimum** number of decks among these $n+1$ that Vasya should take, so that there will be a suitable deck no matter how many girls and boys there will be among the participants of the tournament.

Input

The first line contains an integer b ($1 \le b \le 300$), the number of boys.

The second line contains an integer g ($1 \le g \le 300$), the number of girls.

The third line contains an integer n ($1 \le n \le b+g$), the number of the board games tournament participants.

Output

Output the only integer, the **minimum** number of badge decks that Vasya could take.

Examples

input

```
5
6
3
```

output

```
4
```

input

```
5
3
5
```

output

```
4
```

Note

In the first example, each of 4 decks should be taken: (0 blue, 3 red), (1 blue, 2 red), (2 blue, 1 red), (3 blue, 0 red).

In the second example, 4 decks should be taken: (2 blue, 3 red), (3 blue, 2 red), (4 blue, 1 red), (5 blue, 0 red). Piles (0 blue, 5 red) and (1 blue, 4 red) can not be used.

C. Bad Sequence

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Petya's friends made him a birthday present — a bracket sequence. Petya was quite disappointed with his gift, because he dreamed of correct bracket sequence, yet he told his friends nothing about his dreams and decided to fix present himself.

To make everything right, Petya is going to move at most one bracket from its original place in the sequence to any other position. Reversing the bracket (e.g. turning "(" into ")" or vice versa) isn't allowed.

We remind that bracket sequence s is called correct if:

- s is empty;
- s is equal to "(t)", where t is correct bracket sequence;
- s is equal to $t_1 t_2$, i.e. concatenation of t_1 and t_2, where t_1 and t_2 are correct bracket sequences.

For example, "(()())", "()" are correct, while ")(" and "())" are not. Help Petya to fix his birthday present and understand whether he can move one bracket so that the sequence becomes correct.

Input

First of line of input contains a single number n ($1 \le n \le 200\,000$) — length of the sequence which Petya received for his birthday.

Second line of the input contains bracket sequence of length n, containing symbols "(" and ")".

Output

Print "Yes" if Petya can make his sequence correct moving at most one bracket. Otherwise print "No".

Examples

input

```
2
)(
```

output

```
Yes
```

input

```
3
(()
```

output

```
No
```

input

```
2
()
```

output

```
Yes
```

input

```
10
)))))(((((
```

output

```
No
```

Note

In the first example, Petya can move first bracket to the end, thus turning the sequence into "()", which is correct bracket sequence.

In the second example, there is no way to move at most one bracket so that the sequence becomes correct.

In the third example, the sequence is already correct and there's no need to move brackets.

D. Treasure Island

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

All of us love treasures, right? That's why young Vasya is heading for a Treasure Island.

Treasure Island may be represented as a rectangular table $n \times m$ which is surrounded by the ocean. Let us number rows of the field with consecutive integers from 1 to n from top to bottom and columns with consecutive integers from 1 to m from left to right. Denote the cell in r-th row and c-th column as (r, c). Some of the island cells contain impassable forests, and some cells are free and passable. Treasure is hidden in cell (n, m).

Vasya got off the ship in cell $(1, 1)$. Now he wants to reach the treasure. He is hurrying up, so he can move only from cell to the cell in next row (downwards) or next column (rightwards), i.e. from cell (x, y) he can move only to cells $(x+1, y)$ and $(x, y+1)$. Of course Vasya can't move through cells with impassable forests.

Evil Witch is aware of Vasya's journey and she is going to prevent him from reaching the treasure. Before Vasya's first move she is able to grow using her evil magic impassable forests in previously free cells. Witch is able to grow a forest in any number of any free cells except cells $(1, 1)$ where Vasya got off his ship and (n, m) where the treasure is hidden.

Help Evil Witch by finding out the minimum number of cells she has to turn into impassable forests so that Vasya is no longer able to reach the treasure.

Input

First line of input contains two positive integers n, m ($3 \le n \cdot m \le 1\,000\,000$), sizes of the island.

Following n lines contains strings s_i of length m describing the island, j-th character of string s_i equals "#" if cell (i, j) contains an impassable forest and "." if the cell is free and passable. Let us remind you that Vasya gets of his ship at the cell $(1, 1)$, i.e. the first cell of the first row, and he wants to reach cell (n, m), i.e. the last cell of the last row.

It's guaranteed, that cells $(1, 1)$ and (n, m) are empty.

Output

Print the only integer k, which is the minimum number of cells Evil Witch has to turn into impassable forest in order to prevent Vasya from reaching the treasure.

Examples

input

```
2 2
..
..
```

output

```
2
```

input

```
4 4
....
#.#.
....
.#..
```

output

```
1
```

input

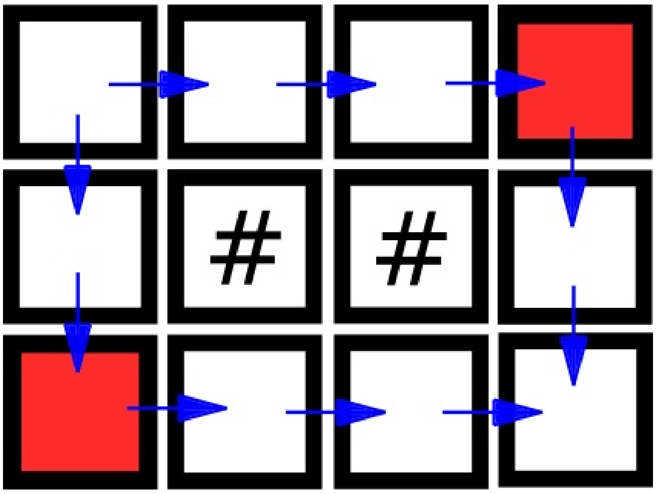
```
3 4
....
.##.
....
```

output

```
2
```

Note

The following picture illustrates the island in the third example. Blue arrows show possible paths Vasya may use to go from $(1, 1)$ to (n, m). Red illustrates one possible set of cells for the Witch to turn into impassable forest to make Vasya's trip from $(1, 1)$ to (n, m) impossible.



E. Petya and Construction Set

time limit per test: 1 second

memory limit per test: 512 megabytes

input: standard input

output: standard output

It's Petya's birthday party and his friends have presented him a brand new "Electrician-n" construction set, which they are sure he will enjoy as he always does with weird puzzles they give him.

Construction set "Electrician-n" consists of $2n - 1$ wires and $2n$ light bulbs. Each bulb has its own unique index that is an integer from 1 to $2n$, while all wires look the same and are indistinguishable. In order to complete this construction set one has to use each of the wires to connect two distinct bulbs. We define a *chain* in a completed construction set as a sequence of distinct bulbs of length at least two, such that every two consecutive bulbs in this sequence are directly connected by a wire. Completed construction set configuration is said to be correct if a resulting network of bulbs and wires has a tree structure, i.e. any two distinct bulbs are the endpoints of some chain.

Petya was assembling different configurations for several days, and he noticed that sometimes some of the bulbs turn on. After a series of experiments he came up with a conclusion that bulbs indexed $2i$ and $2i - 1$ turn on if the chain connecting them consists of exactly d_i wires. Moreover, the following **important** condition holds: the value of d_i is never greater than n.

Petya did his best but was not able to find a configuration that makes all bulbs to turn on, so he seeks your assistance. Please, find out a configuration that makes all bulbs shine. It is guaranteed that such configuration always exists.

Input

The first line of the input contains a single integer n ($1 \le n \le 100\,000$) — the parameter of a construction set that defines the number of bulbs and the number of wires.

Next line contains n integers $d_1, d_2, \ldots, d_n$ ($1 \le d_i \le n$), where d_i stands for the number of wires the chain between bulbs $2i$ and $2i - 1$ should consist of.

Output

Print $2n - 1$ lines. The i-th of them should contain two distinct integers a_i and b_i ($1 \le a_i, b_i \le 2n$, $a_i \ne b_i$) — indices of bulbs connected by a wire.

If there are several possible valid answer you can print any of them.

Examples

input

```
3
2 2 2
```

output

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

input

```
4
2 2 2 1
```

output

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

input

```
6
2 2 2 2 2 2
```

output

```
1 3
2 3
3 5
4 5
5 7
6 7
7 12
8 12
9 11
9 12
10 11
```

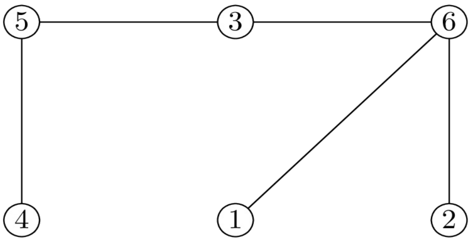
input

```
2
1 1
```

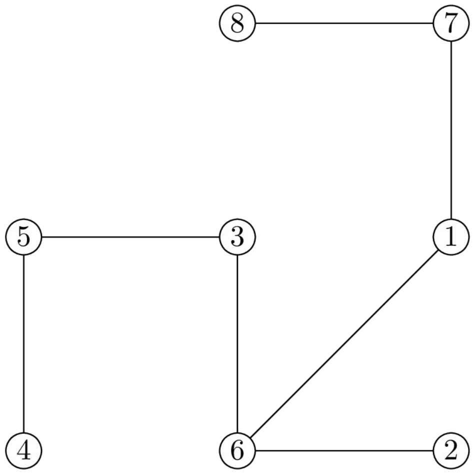
output

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

Note



Answer for the first sample test.



Answer for the second sample test.

F. Employment

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Two large companies "Cecsi" and "Poca Pola" are fighting against each other for a long time. In order to overcome their competitor, "Poca Pola" started a super secret project, for which it has total n vacancies in all of their offices. After many tests and interviews n candidates were selected and the only thing left was their employment.

Because all candidates have the same skills, it doesn't matter where each of them will work. That is why the company decided to distribute candidates between workplaces so that the total distance between home and workplace over all candidates is minimal.

It is well known that Earth is round, so it can be described as a circle, and all m cities on Earth can be described as points on this circle. All cities are enumerated from 1 to m so that for each i $(1 \le i \le m-1)$ cities with indexes i and $i+1$ are neighbors and cities with indexes 1 and m are neighbors as well. People can move only along the circle. The distance between any two cities equals to minimal number of transitions between neighboring cities you have to perform to get from one city to another. In particular, the distance between the city and itself equals 0.

The "Poca Pola" vacancies are located at offices in cities $a_1, a_2, \ldots, a_n$. The candidates live in cities $b_1, b_2, \ldots, b_n$. It is possible that some vacancies are located in the same cities and some candidates live in the same cities.

The "Poca Pola" managers are too busy with super secret project, so you were asked to help "Poca Pola" to distribute candidates between workplaces, so that the sum of the distance between home and workplace over all candidates is minimum possible.

Input

The first line contains two integers m and n $(1 \le m \le 10^9, 1 \le n \le 200\,000)$ — the number of cities on Earth and the number of vacancies.

The second line contains n integers $a_1, a_2, a_3, \ldots, a_n$ $(1 \le a_i \le m)$ — the cities where vacancies are located.

The third line contains n integers $b_1, b_2, b_3, \ldots, b_n$ $(1 \le b_i \le m)$ — the cities where the candidates live.

Output

The first line should contain the minimum total distance between home and workplace over all candidates.

The second line should contain n different integers from 1 to n. The i-th of them should be the index of candidate that should work at i-th workplace.

Examples

input

```
10 3
1 5 5
10 4 6
```

output

```
3
1 2 3
```

input

```
10 3
1 4 8
8 3 6
```

output

```
4
2 3 1
```

Note

In the first example, the distance between each candidate and his workplace equals to 1 (from 1 to 10, from 4 to 5 and from 6 to 5).

In the second example:

- The second candidate works at first workplace, the distance between cities 3 and 1 equals to 2.
- The third candidate works at second workplace, the distance between cities 6 and 4 equals to 2.
- The first candidate works at third workplace, the distance between cities 8 and 8 equals to 0.

G. Feeling Good

time limit per test: 3 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Recently biologists came to a fascinating conclusion about how to find a chameleon mood. Consider chameleon body to be a rectangular table $n \times m$, each cell of which may be green or blue and may change between these two colors. We will denote as (x, y) ($1 \le x \le n$, $1 \le y \le m$) the cell in row x and column y.

Let us define a chameleon *good mood certificate* to be four cells which are corners of some subrectangle of the table, such that colors in opposite cells among these four are similar, and at the same time not all of the four cell colors are similar. Formally, it is a group of four cells (x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2) for some $1 \le x_1 < x_2 \le n$, $1 \le y_1 < y_2 \le m$, that colors of (x_1, y_1) and (x_2, y_2) coincide and colors of (x_1, y_2) and (x_2, y_1) coincide, but not all of the four cells share the same color. It was found that whenever such four cells are present, chameleon is in good mood, and vice versa: if there are no such four cells, chameleon is in bad mood.

You are asked to help scientists write a program determining the mood of chameleon. Let us consider that initially all cells of chameleon are green. After that chameleon coloring may change several times. On one change, colors of contiguous segment of some table row are replaced with the opposite. Formally, each color change is defined by three integers a, l, r ($1 \le a \le n$, $1 \le l \le r \le m$). On such change colors of all cells (a, b) such that $l \le b \le r$ are replaced with the opposite.

Write a program that reports mood of the chameleon after each change. Additionally, if the chameleon mood is good, program should find out any four numbers x_1, y_1, x_2, y_2 such that four cells (x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2) are the good mood certificate.

Input

The first line of input contains three integers n, m, q ($1 \le n, m \le 2000$, $1 \le q \le 500\,000$), the sizes of the table and the number of changes respectively.

Each of the following q lines contains 3 integers a_i, l_i, r_i ($1 \le a_i \le n$, $1 \le l_i \le r_i \le m$), describing i-th coloring change.

Output

Print q lines. In the i-th line report the chameleon mood after first i color changes for all $1 \le i \le q$.

If chameleon is in bad mood, print the only integer -1.

Otherwise, print four integers x_1, y_1, x_2, y_2 ($1 \le x_1 < x_2 \le n$, $1 \le y_1 < y_2 \le m$) such that four cells (x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2) are the good mood certificate. If there are several ways to choose such four integers, print any valid one.

Examples

input

```
2 2 6
1 1 1
2 2 2
2 1 1
1 2 2
2 2 2
1 1 1
```

output

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

input

```
4 3 9
2 2 3
4 1 2
2 1 3
3 2 2
3 1 3
1 2 2
4 2 3
1 1 3
3 1 3
```

output
-1 2 1 4 3 -1 2 1 3 2 3 2 4 3 1 1 2 2 1 1 2 2 -1 2 1 3 2

H. Tiles Placement

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

The new pedestrian zone in Moscow city center consists of n squares connected with each other by $n-1$ footpaths. We define a *simple path* as a sequence of squares such that no square appears in this sequence twice and any two adjacent squares in this sequence are directly connected with a footpath. The size of a simple path is the number of squares in it. The footpaths are designed in a such a way that there is exactly one simple path between any pair of different squares.

During preparations for Moscow City Day the city council decided to renew ground tiles on all n squares. There are k tile types of different colors, numbered from 1 to k. For each square exactly one tile type must be selected and then used to cover this square surface. To make walking through the city center more fascinating, it was decided to select tiles types for each square in such a way that any possible simple path of size exactly k contains squares with all k possible tile colors.

You need to find out whether it is possible to place the tiles this way or not.

Input

The first line contains two integers n, k ($2 \le k \le n \le 200\,000$) — the number of squares in the new pedestrian zone, the number of different tile colors.

Each of the following $n-1$ lines contains two integers v_i and u_i ($1 \le v_i, u_i \le n$) — numbers of the squares connected by the corresponding road.

It's guaranteed, that it's possible to go from any square to any other square, moreover there is exactly one such simple path.

Output

Print "Yes" if it is possible to assign tile colors this way and "No" otherwise.

In case your answer is "Yes", print n integers from 1 to k each, the color of the tile for every square.

Examples

input
7 4 1 3 2 3 3 4 4 5 5 6 5 7

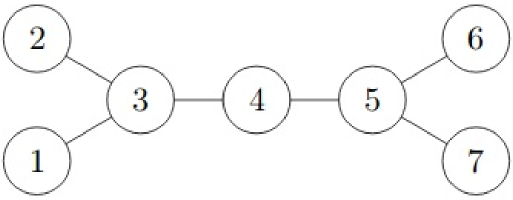
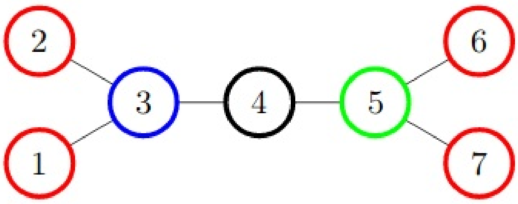
output
Yes 1 1 2 3 4 1 1

input
7 3 1 3 2 3 3 4 4 5 5 6 5 7

output
No

Note

The following pictures illustrate the pedestrian zone in first and second examples. The second picture also shows one possible distribution of colors among the squares for $k = 4$.


1
2
3
4
5
6
7

1
2
3
4
5
6
7