A. Alex and broken contest

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

One day Alex was creating a contest about his friends, but accidentally deleted it. Fortunately, all the problems were saved, but now he needs to find them among other problems.

But there are too many problems, to do it manually. Alex asks you to write a program, which will determine if a problem is from this contest by its name.

It is known, that problem is from this contest if and only if its name contains one of Alex's friends' name **exactly once**. His friends' names are "Danil", "Olya", "Slava", "Ann" and "Nikita".

Names are case sensitive.

Input

The only line contains string from lowercase and uppercase letters and "_" symbols of length, not more than 100 — the name of the problem.

Output

Print "YES", if problem is from this contest, and "NO" otherwise.

Examples

input
```
Alex_and_broken_contest
```
output
```
NO
```

input
```
NikitaAndString
```
output
```
YES
```

input
```
Danil_and_Olya
```
output
```
NO
```

B. Nikita and string

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

One day Nikita found the string containing letters "a" and "b" only.

Nikita thinks that string is beautiful if it can be cut into 3 strings (possibly empty) without changing the order of the letters, where the 1-st and the 3-rd one contain only letters "a" and the 2-nd contains only letters "b".

Nikita wants to make the string beautiful by removing some (possibly none) of its characters, but without changing their order. What is the maximum length of the string he can get?

Input

The first line contains a non-empty string of length not greater than $5\ 000$ containing only lowercase English letters "a" and "b".

Output

Print a single integer — the maximum possible size of beautiful string Nikita can get.

Examples

input

```
abba
```

output

```
4
```

input

```
bab
```

output

```
2
```

Note

It the first sample the string is already beautiful.

In the second sample he needs to delete one of "b" to make it beautiful.

C. Slava and tanks

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

Slava plays his favorite game "Peace Lightning". Now he is flying a bomber on a very specific map.

Formally, map is a checkered field of size $1 \times n$, the cells of which are numbered from 1 to n, in each cell there can be one or several tanks. Slava doesn't know the number of tanks and their positions, because he flies very high, but he can drop a bomb in any cell. All tanks in this cell will be damaged.

If a tank takes damage for the first time, it instantly moves to one of the neighboring cells (a tank in the cell n can only move to the cell $n - 1$, a tank in the cell 1 can only move to the cell 2). If a tank takes damage for the second time, it's counted as destroyed and never moves again. The tanks move only when they are damaged for the first time, they do not move by themselves.

Help Slava to destroy all tanks using as few bombs as possible.

Input

The first line contains a single integer n $(2 \le n \le 100\ 000)$ — the size of the map.

Output

In the first line print m — the minimum number of bombs Slava needs to destroy all tanks.

In the second line print m integers $k_1, k_2, \dots, k_m$. The number k_i means that the i-th bomb should be dropped at the cell k_i.

If there are multiple answers, you can print any of them.

Examples

input

```
2
```

output

```
3
2 1 2
```

input

```
3
```

output

```
4
2 1 3 2
```

D. Olya and Energy Drinks

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

Olya loves energy drinks. She loves them so much that her room is full of empty cans from energy drinks.

Formally, her room can be represented as a field of $n \times m$ cells, each cell of which is empty or littered with cans.

Olya drank a lot of energy drink, so now she can run k meters per second. Each second she chooses one of the four directions (up, down, left or right) and runs from 1 to k meters in this direction. Of course, she can only run through empty cells.

Now Olya needs to get from cell (x_1, y_1) to cell (x_2, y_2). How many seconds will it take her if she moves optimally?

It's guaranteed that cells (x_1, y_1) and (x_2, y_2) are empty. These cells can coincide.

Input

The first line contains three integers n, m and k ($1 \le n, m, k \le 1000$) — the sizes of the room and Olya's speed.

Then n lines follow containing m characters each, the i-th of them contains on j-th position "`#`", if the cell (i, j) is littered with cans, and "`.`" otherwise.

The last line contains four integers x_1, y_1, x_2, y_2 ($1 \le x_1, x_2 \le n$, $1 \le y_1, y_2 \le m$) — the coordinates of the first and the last cells.

Output

Print a single integer — the minimum time it will take Olya to get from (x_1, y_1) to (x_2, y_2).

If it's impossible to get from (x_1, y_1) to (x_2, y_2), print `-1`.

Examples

input

```
3 4 4
....
###.
....
1 1 3 1
```

output

```
3
```

input

```
3 4 1
....
###.
....
1 1 3 1
```

output

```
8
```

input

```
2 2 1
.#
#.
1 1 2 2
```

output

```
-1
```

Note

In the first sample Olya should run 3 meters to the right in the first second, 2 meters down in the second second and 3 meters to the left in the third second.

In second sample Olya should run to the right for 3 seconds, then down for 2 seconds and then to the left for 3 seconds.

Olya does not recommend drinking energy drinks and generally believes that this is bad.

E. Danil and a Part-time Job

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

Danil decided to earn some money, so he had found a part-time job. The interview have went well, so now he is a light switcher.

Danil works in a rooted tree (undirected connected acyclic graph) with n vertices, vertex 1 is the root of the tree. There is a room in each vertex, light can be switched on or off in each room. Danil's duties include switching light in all rooms of the subtree of the vertex. It means that if light is switched on in some room of the subtree, he should switch it off. Otherwise, he should switch it on.

Unfortunately (or fortunately), Danil is very lazy. He knows that his boss is not going to personally check the work. Instead, he will send Danil tasks using `Workforces` personal messages.

There are two types of tasks:

1. `pow v` describes a task to switch lights in the subtree of vertex v.
2. `get v` describes a task to count the number of rooms in the subtree of v, in which the light is turned on. Danil should send the answer to his boss using `Workforces` messages.

A subtree of vertex v is a set of vertices for which the shortest path from them to the root passes through v. In particular, the vertex v is in the subtree of v.

Danil is not going to perform his duties. He asks you to write a program, which answers the boss instead of him.

Input

The first line contains a single integer n $(1 \le n \le 200\ 000)$ — the number of vertices in the tree.

The second line contains $n - 1$ space-separated integers $p_2, p_3, \ldots, p_n$ $(1 \le p_i < i)$, where p_i is the ancestor of vertex i.

The third line contains n space-separated integers $t_1, t_2, \ldots, t_n$ $(0 \le t_i \le 1)$, where t_i is 1, if the light is turned on in vertex i and 0 otherwise.

The fourth line contains a single integer q $(1 \le q \le 200\ 000)$ — the number of tasks.

The next q lines are `get v` or `pow v` $(1 \le v \le n)$ — the tasks described above.

Output

For each task `get v` print the number of rooms in the subtree of v, in which the light is turned on.

Example

input

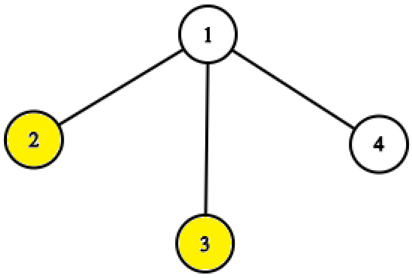
```
4
1 1 1
1 0 0 1
9
get 1
get 2
get 3
get 4
pow 1
get 1
get 2
get 3
get 4
```

output

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

Note

The tree before the task `pow 1`.



The tree after the task `pow 1`.

F. Ann and Books

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

In Ann's favorite book shop are as many as n books on math and economics. Books are numbered from 1 to n. Each of them contains non-negative number of problems.

Today there is a sale: any subsegment of a segment from l to r can be bought at a fixed price.

Ann decided that she wants to buy such non-empty subsegment that the sale operates on it and the number of math problems is greater than the number of economics problems **exactly** by k. Note that k may be positive, negative or zero.

Unfortunately, Ann is not sure on which segment the sale operates, but she has q assumptions. For each of them she wants to know the number of options to buy a subsegment satisfying the condition (because the time she spends on choosing depends on that).

Currently Ann is too busy solving other problems, she asks you for help. For each her assumption determine the number of subsegments of the given segment such that the number of math problems is greaten than the number of economics problems on that subsegment exactly by k.

Input

The first line contains two integers n and k ($1 \le n \le 100\,000$, $-10^9 \le k \le 10^9$) — the number of books and the needed difference between the number of math problems and the number of economics problems.

The second line contains n integers $t_1, t_2, \ldots, t_n$ ($1 \le t_i \le 2$), where t_i is 1 if the i-th book is on math or 2 if the i-th is on economics.

The third line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$), where a_i is the number of problems in the i-th book.

The fourth line contains a single integer q ($1 \le q \le 100\,000$) — the number of assumptions.

Each of the next q lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$) describing the i-th Ann's assumption.

Output

Print q lines, in the i-th of them print the number of subsegments for the i-th Ann's assumption.

Examples

input

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

output

```
2
3
4
1
```

input

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

output

```
10
```

Note

In the first sample Ann can buy subsegments $[1;1]$, $[2;2]$, $[3;3]$, $[2;4]$ if they fall into the sales segment, because the number of math problems is greater by 1 on them that the number of economics problems. So we should count for each assumption the number of these subsegments that are subsegments of the given segment.

Segments $[1;1]$ and $[2;2]$ are subsegments of $[1;2]$.

Segments $[1;1]$, $[2;2]$ and $[3;3]$ are subsegments of $[1;3]$.

Segments $[1;1]$, $[2;2]$, $[3;3]$, $[2;4]$ are subsegments of $[1;4]$.

Segment $[3;3]$ is subsegment of $[3;4]$.