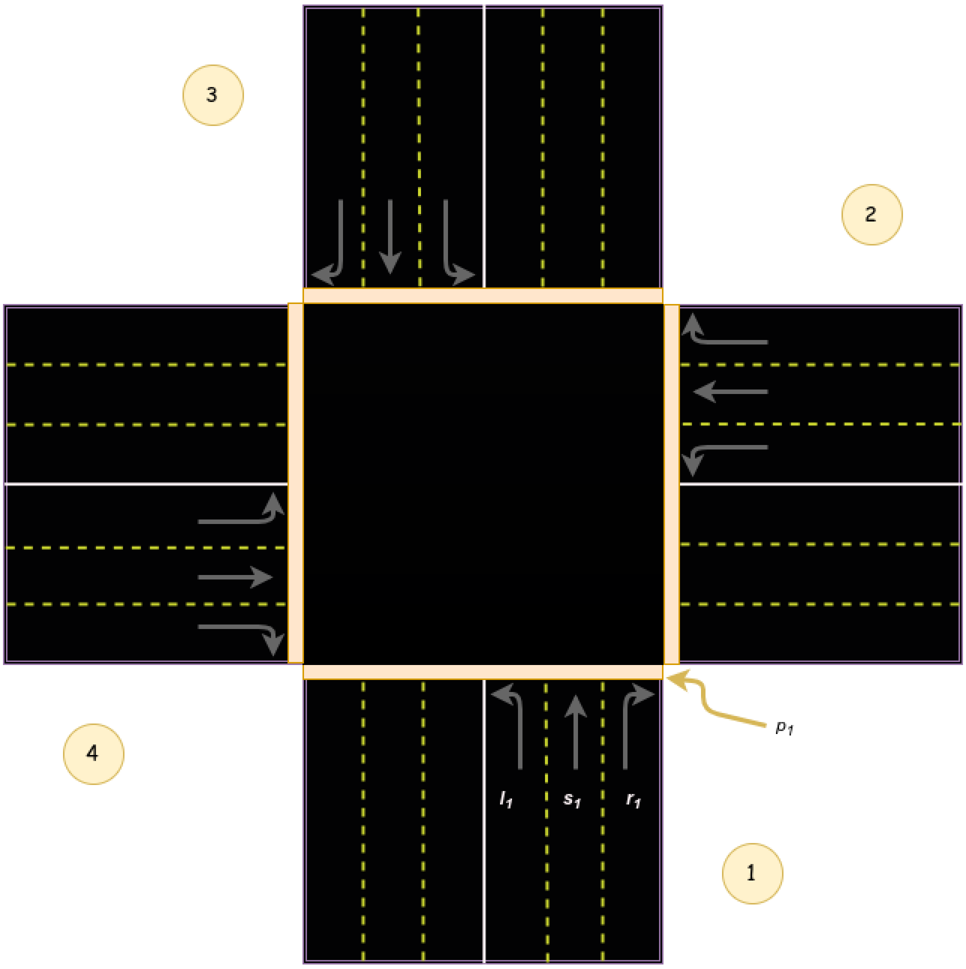
A. Sagheer and Crossroads

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

Sagheer is walking in the street when he comes to an intersection of two roads. Each road can be represented as two parts where each part has 3 lanes getting into the intersection (one for each direction) and 3 lanes getting out of the intersection, so we have 4 parts in total. Each part has 4 lights, one for each lane getting into the intersection (l — left, s — straight, r — right) and a light p for a pedestrian crossing.



An accident is possible if a car can hit a pedestrian. This can happen if the light of a pedestrian crossing of some part and the light of a lane that can get to or from that same part are green at the same time.

Now, Sagheer is monitoring the configuration of the traffic lights. Your task is to help him detect whether an accident is possible.

Input

The input consists of four lines with each line describing a road part given in a counter-clockwise order.

Each line contains four integers l, s, r, p — for the left, straight, right and pedestrian lights, respectively. The possible values are 0 for red light and 1 for green light.

Output

On a single line, print "YES" if an accident is possible, and "NO" otherwise.

Examples

input

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

output

```
YES
```

input

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

output

```
NO
```

input

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

output

```
NO
```

Note

In the first example, some accidents are possible because cars of part 1 can hit pedestrians of parts 1 and 4. Also, cars of parts 2 and 3 can hit pedestrians of part 4.

In the second example, no car can pass the pedestrian crossing of part 4 which is the only green pedestrian light. So, no accident can occur.

B. Sagheer, the Hausmeister

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

Some people leave the lights at their workplaces on when they leave that is a waste of resources. As a hausmeister of DHBW, Sagheer waits till all students and professors leave the university building, then goes and turns all the lights off.

The building consists of n floors with stairs at the left and the right sides. Each floor has m rooms on the same line with a corridor that connects the left and right stairs passing by all the rooms. In other words, the building can be represented as a rectangle with n rows and $m + 2$ columns, where the first and the last columns represent the stairs, and the m columns in the middle represent rooms.

Sagheer is standing at the ground floor at the left stairs. He wants to turn all the lights off in such a way that he will not go upstairs until all lights in the floor he is standing at are off. Of course, Sagheer must visit a room to turn the light there off. It takes one minute for Sagheer to go to the next floor using stairs or to move from the current room/stairs to a neighboring room/stairs on the same floor. It takes no time for him to switch the light off in the room he is currently standing in. Help Sagheer find the minimum total time to turn off all the lights.

Note that Sagheer does not have to go back to his starting position, and he does not have to visit rooms where the light is already switched off.

Input

The first line contains two integers n and m ($1 \le n \le 15$ and $1 \le m \le 100$) — the number of floors and the number of rooms in each floor, respectively.

The next n lines contains the building description. Each line contains a binary string of length $m + 2$ representing a floor (the left stairs, then m rooms, then the right stairs) where 0 indicates that the light is off and 1 indicates that the light is on. The floors are listed from top to bottom, so that the last line represents the ground floor.

The first and last characters of each string represent the left and the right stairs, respectively, so they are always 0.

Output

Print a single integer — the minimum total time needed to turn off all the lights.

Examples

input

```
2 2
0010
0100
```

output

```
5
```

input

```
3 4
001000
000010
000010
```

output

```
12
```

input

```
4 3
01110
01110
01110
01110
```

output

```
18
```

Note

In the first example, Sagheer will go to room 1 in the ground floor, then he will go to room 2 in the second floor using the left or right stairs.

In the second example, he will go to the fourth room in the ground floor, use right stairs, go to the fourth room in the second floor, use right stairs again, then go to the second room in the last floor.

In the third example, he will walk through the whole corridor alternating between the left and right stairs at each floor.

C. Sagheer and Nubian Market

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

On his trip to Luxor and Aswan, Sagheer went to a Nubian market to buy some souvenirs for his friends and relatives. The market has some strange rules. It contains n different items numbered from 1 to n. The i-th item has base cost a_i Egyptian pounds. If Sagheer buys k items with indices $x_1, x_2, \ldots, x_k$, then the cost of item x_j is $a_{x_j} + x_j \cdot k$ for $1 \le j \le k$. In other words, the cost of an item is equal to its base cost in addition to its index multiplied by the factor k.

Sagheer wants to buy as many souvenirs as possible without paying more than S Egyptian pounds. Note that he cannot buy a souvenir more than once. If there are many ways to maximize the number of souvenirs, he will choose the way that will minimize the total cost. Can you help him with this task?

Input

The first line contains two integers n and S ($1 \le n \le 10^5$ and $1 \le S \le 10^9$) — the number of souvenirs in the market and Sagheer's budget.

The second line contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$) — the base costs of the souvenirs.

Output

On a single line, print two integers k, T — the maximum number of souvenirs Sagheer can buy and the minimum total cost to buy these k souvenirs.

Examples

input

```
3 11
2 3 5
```

output

```
2 11
```

input

```
4 100
1 2 5 6
```

output

```
4 54
```

input

```
1 7
7
```

output

```
0 0
```

Note

In the first example, he cannot take the three items because they will cost him $[5, 9, 14]$ with total cost 28. If he decides to take only two items, then the costs will be $[4, 7, 11]$. So he can afford the first and second items.

In the second example, he can buy all items as they will cost him $[5, 10, 17, 22]$.

In the third example, there is only one souvenir in the market which will cost him 8 pounds, so he cannot buy it.

D. Sagheer and Kindergarten

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

Sagheer is working at a kindergarten. There are n children and m different toys. These children use well-defined protocols for playing with the toys:

- Each child has a lovely set of toys that he loves to play with. He requests the toys one after another at distinct moments of time. A child starts playing if and only if he is granted all the toys in his lovely set.
- If a child starts playing, then sooner or later he gives the toys back. No child keeps the toys forever.
- Children request toys at distinct moments of time. No two children request a toy at the same time.
- If a child is granted a toy, he never gives it back until he finishes playing with his lovely set.
- If a child is not granted a toy, he waits until he is granted this toy. He can't request another toy while waiting.
- If two children are waiting for the same toy, then the child who requested it first will take the toy first.

Children don't like to play with each other. That's why they never share toys. When a child requests a toy, then granting the toy to this child depends on whether the toy is free or not. If the toy is free, Sagheer will give it to the child. Otherwise, the child has to wait for it and can't request another toy.

Children are smart and can detect if they have to wait forever before they get the toys they want. In such case they start crying. In other words, a crying set is a set of children in which each child is waiting for a toy that is kept by another child in the set.

Now, we have reached a scenario where all the children made all the requests for their lovely sets, except for one child x that still has one last request for his lovely set. Some children are playing while others are waiting for a toy, but no child is crying, and no one has yet finished playing. If the child x is currently waiting for some toy, he makes his last request just after getting that toy. Otherwise, he makes the request right away. When child x will make his last request, how many children will start crying?

You will be given the scenario and q **independent** queries. Each query will be of the form x y meaning that the last request of the child x is for the toy y. Your task is to help Sagheer find the size of the **maximal** crying set when child x makes his last request.

Input

The first line contains four integers n, m, k, q ($1 \le n, m, k, q \le 10^5$) — the number of children, toys, scenario requests and queries.

Each of the next k lines contains two integers a, b ($1 \le a \le n$ and $1 \le b \le m$) — a scenario request meaning child a requests toy b. The requests are given in the order they are made by children.

Each of the next q lines contains two integers x, y ($1 \le x \le n$ and $1 \le y \le m$) — the request to be added to the scenario meaning child x will request toy y just after getting the toy he is waiting for (if any).

It is guaranteed that the scenario requests are consistent and no child is initially crying. All the scenario requests are distinct and no query coincides with a scenario request.

Output

For each query, print on a single line the number of children who will start crying when child x makes his last request for toy y. Please answer all queries independent of each other.

Examples

input

```
3 3 5 1
1 1
2 2
3 3
1 2
2 3
3 1
```

output

```
3
```

input

```
5 4 7 2
1 1
2 2
2 1
5 1
3 3
4 4
4 1
5 3
5 4
```

output

```
0
2
```

Note

In the first example, child 1 is waiting for toy 2, which child 2 has, while child 2 is waiting for top 3, which child 3 has. When child 3 makes his last request, the toy he requests is held by child 1. Each of the three children is waiting for a toy held by another child and no one is playing, so all the three will start crying.

In the second example, at the beginning, child i is holding toy i for $1 \le i \le 4$. Children 1 and 3 have completed their lovely sets. After they finish playing, toy 3 will be free while toy 1 will be taken by child 2 who has just completed his lovely set. After he finishes, toys 1 and 2 will be free and child 5 will take toy 1. Now:

- In the first query, child 5 will take toy 3 and after he finishes playing, child 4 can play.
- In the second query, child 5 will request toy 4 which is held by child 4. At the same time, child 4 is waiting for toy 1 which is now held by child 5. None of them can play and they will start crying.

E. Sagheer and Apple Tree

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

Sagheer is playing a game with his best friend Soliman. He brought a tree with n nodes numbered from 1 to n and rooted at node 1. The i-th node has a_i apples. This tree has a special property: the lengths of all paths from the root to any leaf have the same parity (i.e. all paths have even length or all paths have odd length).

Sagheer and Soliman will take turns to play. Soliman will make the first move. The player who can't make a move loses.

In each move, the current player will pick a single node, take a non-empty subset of apples from it and do one of the following two things:

1. eat the apples, if the node is a leaf.
2. move the apples to one of the children, if the node is non-leaf.

Before Soliman comes to start playing, Sagheer will make **exactly one change** to the tree. He will pick two different nodes u and v and swap the apples of u with the apples of v.

Can you help Sagheer count the number of ways to make the swap (i.e. to choose u and v) after which he will win the game if both players play optimally? (u, v) and (v, u) are considered to be the same pair.

Input

The first line will contain one integer n $(2 \le n \le 10^5)$ — the number of nodes in the apple tree.

The second line will contain n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^7)$ — the number of apples on each node of the tree.

The third line will contain $n - 1$ integers $p_2, p_3, \ldots, p_n$ $(1 \le p_i \le n)$ — the parent of each node of the tree. Node i has parent p_i (for $2 \le i \le n$). Node 1 is the root of the tree.

It is guaranteed that the input describes a valid tree, and the lengths of all paths from the root to any leaf will have the same parity.

Output

On a single line, print the number of different pairs of nodes (u, v), $u \ne v$ such that if they start playing after swapping the apples of both nodes, Sagheer will win the game. (u, v) and (v, u) are considered to be the same pair.

Examples

input

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

output

```
1
```

input

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

output

```
0
```

input

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

output

```
4
```

Note

In the first sample, Sagheer can only win if he swapped node 1 with node 3. In this case, both leaves will have 2 apples. If Soliman makes a move in a leaf node, Sagheer can make the same move in the other leaf. If Soliman moved some apples from a root to a leaf, Sagheer will eat those moved apples. Eventually, Soliman will not find a move.

In the second sample, There is no swap that will make Sagheer win the game.

Note that Sagheer must make the swap even if he can win with the initial tree.