

Codeforces Round #492 (Div. 1) [Thanks, uDebug!]

A. Tesla

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Allen dreams of one day owning an enormous fleet of electric cars, the car of the future! He knows that this will give him a big status boost. As Allen is planning out all of the different types of cars he will own and how he will arrange them, he realizes that he has a problem.

Allen's future parking lot can be represented as a rectangle with 4 rows and n ($n \leq 50$) columns of rectangular spaces, each of which can contain at most one car at any time. He imagines having k ($k \leq 2n$) cars in the grid, and all the cars are initially in the second and third rows. Each of the cars also has a different designated parking space in the first or fourth row. Allen has to put the cars into corresponding parking places.

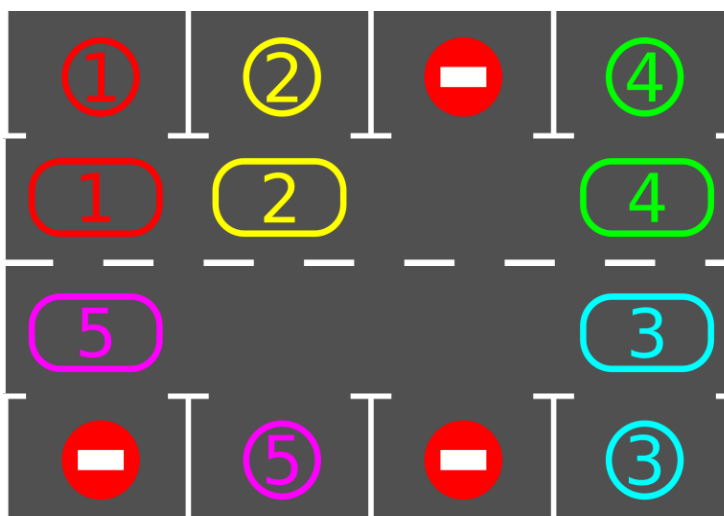


Illustration to the first example.

However, since Allen would never entrust his cars to anyone else, only one car can be moved at a time. He can drive a car from a space in any of the four cardinal directions to a neighboring empty space. Furthermore, Allen can only move one of his cars into a space on the first or fourth rows if it is the car's designated parking space.

Allen knows he will be a very busy man, and will only have time to move cars at most 20000 times before he realizes that moving cars is not worth his time. Help Allen determine if he should bother parking his cars or leave it to someone less important.

Input

The first line of the input contains two space-separated integers n and k ($1 \leq n \leq 50$, $1 \leq k \leq 2n$), representing the number of columns and the number of cars, respectively.

The next four lines will contain n integers each between 0 and k inclusive, representing the initial state of the parking lot. The rows are numbered 1 to 4 from top to bottom and the columns are numbered 1 to n from left to right.

In the first and last line, an integer x represents a parking spot assigned to car x (you can only move this car to this place), while the integer 0 represents an empty space (you can't move any car to this place).

In the second and third line, an integer x represents initial position of car x , while the integer 0 represents an empty space (you can move any car to this place).

Each x between 1 and k appears exactly once in the second and third line, and exactly once in the first and fourth line.

Output

If there is a sequence of moves that brings all of the cars to their parking spaces, with at most 20000 car moves, then print m , the number of moves, on the first line. On the following m lines, print the moves (one move per line) in the format $i\ r\ c$, which corresponds to Allen moving car i to the neighboring space at row r and column c .

If it is not possible for Allen to move all the cars to the correct spaces with at most 20000 car moves, print a single line with the integer -1.

Examples

input

```
4 5
1 2 0 4
```

| |
|---|
| 1 2 0 4 5 0 0 3 0 5 0 3 |
| output |
| 6 1 1 1 2 1 2 4 1 4 3 4 4 5 3 2 5 4 2 |

| |
|-------------------------|
| input |
| 1 2 1 2 1 2 |
| output |
| -1 |

| |
|-------------------------|
| input |
| 1 2 1 1 2 2 |
| output |
| 2 1 1 1 2 4 1 |

Note

In the first sample test case, all cars are in front of their spots except car \$\$\$5\$\$\$, which is in front of the parking spot adjacent. The example shows the shortest possible sequence of moves, but any sequence of length at most \$\$\$20000\$\$\$ will be accepted.

In the second sample test case, there is only one column, and the cars are in the wrong order, so no cars can move and the task is impossible.

B. Suit and Tie

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

Allen is hosting a formal dinner party. \$\$\$2n\$\$\$ people come to the event in \$\$\$n\$\$\$ pairs (couples). After a night of fun, Allen wants to line everyone up for a final picture. The \$\$\$2n\$\$\$ people line up, but Allen doesn't like the ordering. Allen prefers if each pair occupies adjacent positions in the line, as this makes the picture more aesthetic.

Help Allen find the minimum number of swaps of **adjacent** positions he must perform to make it so that each couple occupies adjacent positions in the line.

Input

The first line contains a single integer \$\$\$n\$\$\$ (\$\$\$1 \le n \le 100\$\$\$), the number of pairs of people.

The second line contains \$\$\$2n\$\$\$ integers \$\$\$a_1, a_2, \dots, a_{2n}\$\$\$. For each \$\$\$i\$\$\$ with \$\$\$1 \le i \le n\$\$\$, \$\$\$i\$\$\$ appears exactly twice. If \$\$\$a_j = a_k = i\$\$\$, that means that the \$\$\$j\$\$\$-th and \$\$\$k\$\$\$-th people in the line form a couple.

Output

Output a single integer, representing the minimum number of adjacent swaps needed to line the people up so that each pair occupies adjacent positions.

| |
|----------------------|
| Examples |
| input |
| 4 1 1 2 3 3 2 4 4 |
| output |
| 2 |

| |
|------------------|
| input |
| 3 1 1 2 2 3 3 |
| output |

| |
|------------------|
| 0 |
| input |
| 3 3 1 2 3 1 2 |
| output |
| 3 |

Note

In the first sample case, we can transform \$\$\$1 1 2 3 3 2 4 4 \rightarrow 1 1 2 3 2 3 4 4 \rightarrow 1 1 2 2 3 3 4 4\$\$\$ in two steps. Note that the sequence \$\$\$1 1 2 3 3 2 4 4 \rightarrow 1 1 3 2 3 2 4 4 \rightarrow 1 1 3 3 2 2 4 4\$\$\$ also works in the same number of steps.

The second sample case already satisfies the constraints; therefore we need \$\$\$0\$\$\$ swaps.

C. Leaving the Bar

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

For a vector $\vec{v} = (x, y)$, define $|\vec{v}| = \sqrt{x^2 + y^2}$.

Allen had a bit too much to drink at the bar, which is at the origin. There are n vectors $\vec{v_1}, \vec{v_2}, \ldots, \vec{v_n}$. Allen will make n moves. As Allen's sense of direction is impaired, during the i -th move he will either move in the direction $\vec{v_i}$ or $-\vec{v_i}$. In other words, if his position is currently $p = (x, y)$, he will either move to $p + \vec{v_i}$ or $p - \vec{v_i}$.

Allen doesn't want to wander too far from home (which happens to also be the bar). You need to help him figure out a sequence of moves (a sequence of signs for the vectors) such that his final position p satisfies $|p| \leq 1.5 \cdot 10^6$ so that he can stay safe.

Input

The first line contains a single integer n ($1 \leq n \leq 10^5$) — the number of moves.

Each of the following lines contains two space-separated integers x_i and y_i , meaning that $\vec{v_i} = (x_i, y_i)$. We have that $|v_i| \leq 10^6$ for all i .

Output

Output a single line containing n integers c_1, c_2, \ldots, c_n , each of which is either 1 or -1 . Your solution is correct if the value of $p = \sum_{i = 1}^n c_i \vec{v_i}$, satisfies $|p| \leq 1.5 \cdot 10^6$.

It can be shown that a solution always exists under the given constraints.

Examples

| |
|---------------------------------------|
| input |
| 3 999999 0 0 999999 999999 0 |
| output |
| 1 1 -1 |

| |
|---------------------|
| input |
| 1 -824590 246031 |
| output |
| 1 |

| |
|--|
| input |
| 8 -67761 603277 640586 -396671 46147 -122580 569609 -2112 400 914208 131792 309779 -850150 -486293 5272 721899 |
| output |
| 1 1 1 1 1 1 1 -1 |

D. Game

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Allen and Bessie are playing a simple number game. They both know a function $f: \{0, 1\}^n \rightarrow \mathbb{R}$, i. e. the function takes n binary arguments and returns a real value. At the start of the game, the variables x_1, x_2, \dots, x_n are all set to -1 . Each round, with equal probability, one of Allen or Bessie gets to make a move. A move consists of picking an x_i such that $x_i = -1$ and either setting x_i to 0 or x_i to 1 .

After n rounds all variables are set, and the game value resolves to $f(x_1, x_2, \dots, x_n)$. Allen wants to maximize the game value, and Bessie wants to minimize it.

Your goal is to help Allen and Bessie find the expected game value! They will play $r+1$ times though, so between each game, exactly one value of f changes. In other words, between rounds i and $i+1$ for $1 \leq i \leq r$, $f(z_1, \dots, z_n) \rightarrow g_i$ for some $(z_1, \dots, z_n) \in \{0, 1\}^n$. You are to find the expected game value in the beginning and after each change.

Input

The first line contains two integers n and r ($1 \leq n \leq 18$, $0 \leq r \leq 2^{18}$).

The next line contains 2^n integers $c_0, c_1, \dots, c_{2^n-1}$ ($0 \leq c_i \leq 10^9$), denoting the initial values of f . More specifically, $f(x_0, x_1, \dots, x_{n-1}) = c_x$, if $x = \overline{x_{n-1} \dots x_0}$ in binary.

Each of the next r lines contains two integers z and g ($0 \leq z \leq 2^n - 1$, $0 \leq g \leq 10^9$). If $z = \overline{z_{n-1} \dots z_0}$ in binary, then this means to set $f(z_0, \dots, z_{n-1}) \rightarrow g$.

Output

Print $r+1$ lines, the i -th of which denotes the value of the game f during the i -th round. Your answer must have absolute or relative error within 10^{-6} .

Formally, let your answer be a , and the jury's answer be b . Your answer is considered correct if $\frac{|a - b|}{\max\{1, |b|\}} \leq 10^{-6}$.

Examples

| |
|----------------------------------|
| input |
| 2 2 0 1 2 3 2 5 0 4 |
| output |
| 1.500000 2.250000 3.250000 |

| |
|---------------|
| input |
| 1 0 2 3 |
| output |
| 2.500000 |

| |
|----------------|
| input |
| 2 0 1 1 1 1 |
| output |
| 1.000000 |

Note

Consider the second test case. If Allen goes first, he will set x_1 to 1 , so the final value will be 3 . If Bessie goes first, then she will set x_1 to 0 so the final value will be 2 . Thus the answer is 2.5 .

In the third test case, the game value will always be 1 regardless of Allen and Bessie's play.

E. Number Clicker

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Allen is playing Number Clicker on his phone.

He starts with an integer u on the screen. Every second, he can press one of 3 buttons.

- 1. Turn $u \rightarrow u+1 \pmod{p}$.
- 2. Turn $u \rightarrow u+p-1 \pmod{p}$.
- 3. Turn $u \rightarrow u^{p-2} \pmod{p}$.

Allen wants to press at most 200 buttons and end up with v on the screen. Help him!

Input

The first line of the input contains 3 positive integers: u, v, p ($0 \leq u, v \leq p-1$, $3 \leq p \leq 10^9 + 9$). p is guaranteed to be prime.

Output

On the first line, print a single integer ℓ , the number of button presses. On the second line, print integers c_1, \dots, c_ℓ , the button presses. For $1 \leq i \leq \ell$, $c_i \leq 3$.

We can show that the answer always exists.

Examples

| input |
|----------|
| 1 3 5 |
| output |
| 2 1 1 |

| input |
|--------|
| 3 2 5 |
| output |
| 1 3 |

Note

In the first example the integer on the screen changes as $1 \rightarrow 2 \rightarrow 3$.

In the second example the integer on the screen changes as $3 \rightarrow 2$.

F. Cowmpany Cowmpensation

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

Allen, having graduated from the MOO Institute of Techcowlogy (MIT), has started a startup! Allen is the president of his startup. He also hires $n-1$ other employees, each of which is assigned a direct superior. If u is a superior of v and v is a superior of w then also u is a superior of w . Additionally, there are no u and v such that u is the superior of v and v is the superior of u . Allen himself has no superior. Allen is employee number 1 , and the others are employee numbers 2 through n .

Finally, Allen must assign salaries to each employee in the company including himself. Due to budget constraints, each employee's salary is an integer between 1 and D . Additionally, no employee can make strictly more than his superior.

Help Allen find the number of ways to assign salaries. As this number may be large, output it modulo $10^9 + 7$.

Input

The first line of the input contains two integers n and D ($1 \leq n \leq 3000$, $1 \leq D \leq 10^9$).

The remaining $n-1$ lines each contain a single positive integer, where the i -th line contains the integer p_i ($1 \leq p_i \leq i$). p_i denotes the direct superior of employee $i+1$.

Output

Output a single integer: the number of ways to assign salaries modulo $10^9 + 7$.

Examples

| input |
|---------------|
| 3 2 1 1 |
| output |
| 5 |

| input |
|----------|
| 3 3 1 |

| |
|--------|
| 2 |
| output |
| 10 |
| input |
| 2 5 |
| 1 |
| output |
| 15 |

Note
 In the first sample case, employee 2 and 3 report directly to Allen. The three salaries, in order, can be \$\$\$\$(1,1,1)\$\$\$, \$\$\$\$(2,1,1)\$\$\$, \$\$\$\$(2,1,2)\$\$\$, \$\$\$\$(2,2,1)\$\$\$ or \$\$\$\$(2,2,2)\$\$\$.

In the second sample case, employee 2 reports to Allen and employee 3 reports to employee 2. In order, the possible salaries are \$\$\$\$(1,1,1)\$\$\$, \$\$\$\$(2,1,1)\$\$\$, \$\$\$\$(2,2,1)\$\$\$, \$\$\$\$(2,2,2)\$\$\$, \$\$\$\$(3,1,1)\$\$\$, \$\$\$\$(3,2,1)\$\$\$, \$\$\$\$(3,2,2)\$\$\$, \$\$\$\$(3,3,1)\$\$\$, \$\$\$\$(3,3,2)\$\$\$, \$\$\$\$(3,3,3)\$\$\$.