Codeforces Round #203 (Div. 2)

A. TL

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

Valera wanted to prepare a Codesecrof round. He's already got one problem and he wants to set a time limit (TL) on it.

Valera has written n correct solutions. For each correct solution, he knows its running time (in seconds). Valera has also wrote m wrong solutions and for each wrong solution he knows its running time (in seconds).

Let's suppose that Valera will set v seconds TL in the problem. Then we can say that a solution passes the system testing if its running time is at most v seconds. We can also say that a solution passes the system testing with some "extra" time if for its running time, a seconds, an inequality $2a \le v$ holds.

As a result, Valera decided to set v seconds TL, that the following conditions are met:

1. v is a positive integer;
2. all correct solutions pass the system testing;
3. at least one correct solution passes the system testing with some "extra" time;
4. all wrong solutions do not pass the system testing;
5. value v is minimum among all TLs, for which points $1, 2, 3, 4$ hold.

Help Valera and find the most suitable TL or else state that such TL doesn't exist.

Input

The first line contains two integers n, m ($1 \le n, m \le 100$). The second line contains n space-separated positive integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 100$) — the running time of each of the n correct solutions in seconds. The third line contains m space-separated positive integers $b_1, b_2, ..., b_m$ ($1 \le b_i \le 100$) — the running time of each of m wrong solutions in seconds.

Output

If there is a valid TL value, print it. Otherwise, print -1.

Examples

input
3 6 4 5 2 8 9 6 10 7 11
output
5

input
3 1 3 4 5 6
output
-1

B. Resort

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

Valera's finally decided to go on holiday! He packed up and headed for a ski resort.

Valera's fancied a ski trip but he soon realized that he could get lost in this new place. Somebody gave him a useful hint: the resort has n objects (we will consider the objects indexed in some way by integers from 1 to n), each object is either a hotel or a mountain.

Valera has also found out that the ski resort had multiple ski tracks. Specifically, for each object v, the resort has at most one object u, such that there is a ski track built from object u to object v. We also know that no hotel has got a ski track leading from the hotel to some object.

Valera is afraid of getting lost on the resort. So he wants you to come up with a path he would walk along. The path must consist of objects $v_1, v_2, \ldots, v_k$ ($k \geq 1$) and meet the following conditions:

1. Objects with numbers $v_1, v_2, \ldots, v_{k-1}$ are mountains and the object with number v_k is the hotel.
2. For any integer i ($1 \leq i < k$), there is **exactly one** ski track leading from object v_i. This track goes to object v_{i+1}.
3. The path contains as many objects as possible (k is maximal).

Help Valera. Find such path that meets all the criteria of our hero!

Input

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

The second line contains n space-separated integers $type_1, type_2, \ldots, type_n$ — the types of the objects. If $type_i$ equals zero, then the i-th object is the mountain. If $type_i$ equals one, then the i-th object is the hotel. It is guaranteed that at least one object is a hotel.

The third line of the input contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq n$) — the description of the ski tracks. If number a_i equals zero, then there is no such object v, that has a ski track built from v to i. If number a_i doesn't equal zero, that means that there is a track built from object a_i to object i.

Output

In the first line print k — the maximum possible path length for Valera. In the second line print k integers $v_1, v_2, \ldots, v_k$ — the path. If there are multiple solutions, you can print any of them.

Examples

input
5 0 0 0 0 1 0 1 2 3 4
output
5 1 2 3 4 5

input
5 0 0 1 0 1 0 1 2 2 4
output
2 4 5

input
4 1 0 0 0 2 3 4 2
output
1 1

C. Bombs

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

You've got a robot, its task is destroying bombs on a square plane. Specifically, the square plane contains n bombs, the i-th bomb is at point with coordinates (x_i, y_i). We know that no two bombs are at the same point and that no bomb is at point with coordinates $(0, 0)$. Initially, the robot is at point with coordinates $(0, 0)$. Also, let's mark the robot's current position as (x, y). In order to destroy all the bombs, the robot can perform three types of operations:

1. Operation has format "1 k dir". To perform the operation robot have to move in direction dir k ($k \geq 1$) times. There are only 4 directions the robot can move in: "R", "L", "U", "D". During one move the robot can move from the current point to one of following points: $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$ (corresponding to directions). It is forbidden to move from point (x, y), if at least one point on the path (besides the destination point) contains a bomb.
2. Operation has format "2". To perform the operation robot have to pick a bomb at point (x, y) and put it in a special container. Thus, the robot can carry the bomb from any point to any other point. The operation cannot be performed if point (x, y) has no bomb. It is forbidden to pick a bomb if the robot already has a bomb in its container.
3. Operation has format "3". To perform the operation robot have to take a bomb out of the container and destroy it. You are allowed to perform this operation only if the robot is at point $(0, 0)$. It is forbidden to perform the operation if the container has no bomb.

Help the robot and find the shortest possible sequence of operations he can perform to destroy all bombs on the coordinate plane.

Input

The first line contains a single integer n ($1 \leq n \leq 10^5$) — the number of bombs on the coordinate plane. Next n lines contain two integers each. The i-th line contains numbers (x_i, y_i) ($-10^9 \leq x_i, y_i \leq 10^9$) — the coordinates of the i-th bomb. It is guaranteed that no two bombs are located at the same point and no bomb is at point $(0, 0)$.

Output

In a single line print a single integer k — the minimum number of operations needed to destroy all bombs. On the next lines print the descriptions of these k operations. If there are multiple sequences, you can print any of them. It is guaranteed that there is the solution where $k \leq 10^6$.

Examples

input
2 1 1 -1 -1
output
12 1 1 R 1 1 U 2 1 1 L 1 1 D 3 1 1 L 1 1 D 2 1 1 R 1 1 U 3

input
3 5 0 0 5 1 0
output
12 1 1 R 2 1 1 L 3 1 5 R 2 1 5 L 3 1 5 U 2

```
1 5 D
3
```

D. Looking for Owls

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

Emperor Palpatine loves owls very much. The emperor has some blueprints with the new Death Star, the blueprints contain n distinct segments and m distinct circles. We will consider the segments indexed from 1 to n in some way and the circles — indexed from 1 to m in some way.

Palpatine defines an owl as a set of a pair of distinct circles (i, j) ($i < j$) and one segment k, such that:

1. circles i and j are symmetrical relatively to the straight line containing segment k;
2. circles i and j don't have any common points;
3. circles i and j have the same radius;
4. segment k intersects the segment that connects the centers of circles i and j.

Help Palpatine, count the number of distinct owls on the picture.

Input

The first line contains two integers — n and m ($1 \le n \le 3 \cdot 10^5$, $2 \le m \le 1500$).

The next n lines contain four integers each, x_1, y_1, x_2, y_2 — the coordinates of the two endpoints of the segment. It's guaranteed that each segment has positive length.

The next m lines contain three integers each, x_i, y_i, r_i — the coordinates of the center and the radius of the i-th circle. All coordinates are integers of at most 10^4 in their absolute value. The radius is a positive integer of at most 10^4.

It is guaranteed that all segments and all circles are dictinct.

Output

Print a single number — the answer to the problem.

Please, do not use the `%lld` specifier to output 64-bit integers is C++. It is preferred to use the `cout` stream or the `%I64d` specifier.

Examples

input
1 2 3 2 3 -2 0 0 2 6 0 2
output
1

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

input
1 2 -1 0 1 0 -100 0 1 100 0 1
output
0

Note

Here's an owl from the first sample. The owl is sitting and waiting for you to count it.

E. Wrong Floyd

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

Valera conducts experiments with algorithms that search for shortest paths. He has recently studied the Floyd's algorithm, so it's time to work with it.

Valera's already written the code that counts the shortest distance between any pair of vertexes in a **non-directed connected graph** from n vertexes and m edges, containing no loops and multiple edges. Besides, Valera's decided to mark part of the vertexes. He's marked exactly k vertexes $a_1, a_2, \ldots, a_k$.

Valera's code is given below.

```
ans[i][j] // the shortest distance for a pair of vertexes i, j
a[i]  // vertexes, marked by Valera

for(i = 1; i <= n; i++) {
   for(j = 1; j <= n; j++) {
      if (i == j)
         ans[i][j] = 0;
      else
         ans[i][j] = INF;  //INF is a very large number
   }
}

for(i = 1; i <= m; i++) {
   read a pair of vertexes u, v that have a non-directed edge between them;
   ans[u][v] = 1;
   ans[v][u] = 1;
}

for (i = 1; i <= k; i++) {
   v = a[i];
   for(j = 1; j <= n; j++)
      for(r = 1; r <= n; r++)
         ans[j][r] = min(ans[j][r], ans[j][v] + ans[v][r]);
}
```

Valera has seen that his code is wrong. Help the boy. Given the set of marked vertexes $a_1, a_2, \ldots, a_k$, find such **non-directed connected graph**, consisting of n vertexes and m edges, for which Valera's code counts the wrong shortest distance for at least one pair of vertexes (i, j). Valera is really keen to get a graph without any loops and multiple edges. If no such graph exists, print -1.

Input

The first line of the input contains three integers n, m, k ($3 \le n \le 300$, $2 \le k \le n$, $n-1 \le m \le \frac{n(n-1)}{2}$) — the number of vertexes, the number of edges and the number of marked vertexes.

The second line of the input contains k space-separated integers $a_1, a_2, \ldots a_k$ ($1 \le a_i \le n$) — the numbers of the marked vertexes. It is guaranteed that all numbers a_i are distinct.

Output

If the graph doesn't exist, print -1 on a single line. Otherwise, print m lines, each containing two integers u, v — the description of the edges of the graph Valera's been looking for.

Examples

input
3 2 2 1 2
output
1 3 2 3

input
3 3 2 1 2

output
-1