Lista AA 2022 #5

A. Restore Graph

1 second, 256 megabytes

Valera had an undirected connected graph without self-loops and multiple edges consisting of n vertices. The graph had an interesting property: there were at most k edges adjacent to each of its vertices. For convenience, we will assume that the graph vertices were indexed by integers from 1 to n.

One day Valera counted the shortest distances from one of the graph vertices to all other ones and wrote them out in array d. Thus, element $d[i]$ of the array shows the shortest distance from the vertex Valera chose to vertex number i.

Then something irreparable terrible happened. Valera lost the initial graph. However, he still has the array d. Help him restore the lost graph.

Input

The first line contains two space-separated integers n and k $(1 \le k < n \le 10^5)$. Number n shows the number of vertices in the original graph. Number k shows that at most k edges were adjacent to each vertex in the original graph.

The second line contains space-separated integers $d[1], d[2], ..., d[n]$ $(0 \le d[i] < n)$. Number $d[i]$ shows the shortest distance from the vertex Valera chose to the vertex number i.

Output

If Valera made a mistake in his notes and the required graph doesn't exist, print in the first line number -1. Otherwise, in the first line print integer m $(0 \le m \le 10^6)$ — the number of edges in the found graph.

In each of the next m lines print two space-separated integers a_i and b_i $(1 \le a_i, b_i \le n;\ a_i \ne b_i)$, denoting the edge that connects vertices with numbers a_i and b_i. The graph shouldn't contain self-loops and multiple edges. If there are multiple possible answers, print any of them.

input
```
3 2
0 1 1
```
output
```
3
1 2
1 3
3 2
```

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

input
```
3 1
0 0 0
```
output
```
-1
```

B. Border

1 second, 256 megabytes

Astronaut Natasha arrived on Mars. She knows that the Martians are very poor aliens. To ensure a better life for the Mars citizens, their emperor decided to take tax from every tourist who visited the planet. Natasha is the inhabitant of Earth, therefore she had to pay the tax to enter the territory of Mars.

There are n banknote denominations on Mars: the value of i-th banknote is a_i. Natasha has an infinite number of banknotes of each denomination.

Martians have k fingers on their hands, so they use a number system with base k. In addition, the Martians consider the digit d (in the number system with base k) divine. Thus, if the last digit in Natasha's tax amount written in the number system with the base k is d, the Martians will be happy. Unfortunately, Natasha does not know the Martians' divine digit yet.

Determine for which values d Natasha can make the Martians happy.

Natasha can use only her banknotes. Martians don't give her change.

Input

The first line contains two integers n and k $(1 \le n \le 100\,000,$ $2 \le k \le 100\,000)$ — the number of denominations of banknotes and the base of the number system on Mars.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$ — denominations of banknotes on Mars.

All numbers are given in decimal notation.

Output

On the first line output the number of values d for which Natasha can make the Martians happy.

In the second line, output all these values in increasing order.

Print all numbers in decimal notation.

input
```
2 8
12 20
```
output
```
2
0 4
```

input
```
3 10
10 20 30
```
output
```
1
0
```

Consider the first test case. It uses the octal number system.

If you take one banknote with the value of 12, you will get 14_8 in octal system. The last digit is 4_8.

If you take one banknote with the value of 12 and one banknote with the value of 20, the total value will be 32. In the octal system, it is 40_8. The last digit is 0_8.

If you take two banknotes with the value of 20, the total value will be 40, this is 50_8 in the octal system. The last digit is 0_8.

No other digits other than 0_8 and 4_8 can be obtained. Digits 0_8 and 4_8 could also be obtained in other ways.

The second test case uses the decimal number system. The nominals of all banknotes end with zero, so Natasha can give the Martians only the amount whose decimal notation also ends with zero.

C. A Twisty Movement

1 second, 256 megabytes

A dragon symbolizes wisdom, power and wealth. On Lunar New Year's Day, people model a dragon with bamboo strips and clothes, raise them with rods, and hold the rods high and low to resemble a flying dragon.

A performer holding the rod low is represented by a 1, while one holding it high is represented by a 2. Thus, the line of performers can be represented by a sequence $a_1, a_2, ..., a_n$.

Little Tommy is among them. He would like to choose an interval $[l, r]$ ($1 \le l \le r \le n$), then reverse $a_l, a_{l+1}, ..., a_r$ so that the length of the longest non-decreasing subsequence of the new sequence is maximum.

A non-decreasing subsequence is a sequence of indices $p_1, p_2, ..., p_k$, such that $p_1 < p_2 < ... < p_k$ and $a_{p_1} \le a_{p_2} \le ... \le a_{p_k}$. The length of the subsequence is k.

Input

The first line contains an integer n $(1 \le n \le 2000)$, denoting the length of the original sequence.

The second line contains n space-separated integers, describing the original sequence $a_1, a_2, ..., a_n$ $(1 \le a_i \le 2, i = 1, 2, ..., n)$.

Output

Print a single integer, which means the maximum possible length of the longest non-decreasing subsequence of the new sequence.

input
```
4
1 2 1 2
```
output
```
4
```

input
```
10
1 1 2 2 2 1 1 2 2 1
```
output
```
9
```

In the first example, after reversing $[2, 3]$, the array will become $[1, 1, 2, 2]$, where the length of the longest non-decreasing subsequence is 4.

In the second example, after reversing $[3, 7]$, the array will become $[1, 1, 1, 1, 2, 2, 2, 2, 2, 1]$, where the length of the longest non-decreasing subsequence is 9.

D. Restore Permutation

2 seconds, 256 megabytes

An array of integers $p_1, p_2, \ldots, p_n$ is called a permutation if it contains each number from 1 to n exactly once. For example, the following arrays are permutations: $[3, 1, 2]$, $[1]$, $[1, 2, 3, 4, 5]$ and $[4, 3, 1, 2]$. The following arrays are not permutations: $[2]$, $[1, 1]$, $[2, 3, 4]$.

There is a hidden permutation of length n.

For each index i, you are given s_i, which equals to the sum of all p_j such that $j < i$ and $p_j < p_i$. In other words, s_i is the sum of elements before the i-th element that are smaller than the i-th element.

Your task is to restore the permutation.

Input

The first line contains a single integer n $(1 \le n \le 2 \cdot 10^5)$ — the size of the permutation.

The second line contains n integers $s_1, s_2, \ldots, s_n$ $(0 \le s_i \le \frac{n(n-1)}{2})$.

It is guaranteed that the array s corresponds to a valid permutation of length n.

Output

Print n integers $p_1, p_2, \ldots, p_n$ — the elements of the restored permutation. We can show that the answer is always unique.

input
```
3
0 0 0
```
output
```
3 2 1
```

input
```
2
0 1
```
output
```
1 2
```

input
```
5
0 1 1 1 10
```
output
```
1 4 3 2 5
```

In the first example for each i there is no index j satisfying both conditions, hence s_i are always 0.

In the second example for $i = 2$ it happens that $j = 1$ satisfies the conditions, so $s_2 = p_1$.

In the third example for $i = 2, 3, 4$ only $j = 1$ satisfies the conditions, so $s_2 = s_3 = s_4 = 1$. For $i = 5$ all $j = 1, 2, 3, 4$ are possible, so $s_5 = p_1 + p_2 + p_3 + p_4 = 10$.

E. Zero Tree

2 seconds, 256 megabytes

A *tree* is a graph with n vertices and exactly $n - 1$ edges; this graph should meet the following condition: there exists exactly one shortest (by number of edges) path between any pair of its vertices.

A *subtree* of a tree T is a tree with both vertices and edges as subsets of vertices and edges of T.

You're given a tree with n vertices. Consider its vertices numbered with integers from 1 to n. Additionally an integer is written on every vertex of this tree. Initially the integer written on the i-th vertex is equal to v_i. In one move you can apply the following operation:

1. Select the subtree of the given tree that includes the vertex with number 1.
2. Increase (or decrease) by one all the integers which are written on the vertices of that subtree.

Calculate the minimum number of moves that is required to make all the integers written on the vertices of the given tree equal to zero.

Input

The first line of the input contains n $(1 \le n \le 10^5)$. Each of the next $n - 1$ lines contains two integers a_i and b_i $(1 \le a_i, b_i \le n; a_i \ne b_i)$ indicating there's an edge between vertices a_i and b_i. It's guaranteed that the input graph is a tree.

The last line of the input contains a list of n space-separated integers $v_1, v_2, ..., v_n$ $(|v_i| \le 10^9)$.

Output

Print the minimum number of operations needed to solve the task.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

input
```
3
1 2
1 3
1 -1 1
```
output
```
3
```

F. Too Easy Problems

2 seconds, 256 megabytes

You are preparing for an exam on scheduling theory. The exam will last for exactly T milliseconds and will consist of n problems. You can either solve problem i in exactly t_i milliseconds or ignore it and spend no time. You don't need time to rest after solving a problem, either.

Unfortunately, your teacher considers some of the problems too easy for you. Thus, he assigned an integer a_i to every problem i meaning that the problem i can bring you a point to the final score only in case you have solved no more than a_i problems overall (including problem i).

Formally, suppose you solve problems $p_1, p_2, ..., p_k$ during the exam. Then, your final score s will be equal to the number of values of j between 1 and k such that $k \le a_{p_j}$.

You have guessed that the real first problem of the exam is already in front of you. Therefore, you want to choose a set of problems to solve during the exam maximizing your final score in advance. Don't forget that the exam is limited in time, and you must have enough time to solve all chosen problems. If there exist different sets of problems leading to the maximum final score, any of them will do.

Input

The first line contains two integers n and T ($1 \le n \le 2 \cdot 10^5$; $1 \le T \le 10^9$) — the number of problems in the exam and the length of the exam in milliseconds, respectively.

Each of the next n lines contains two integers a_i and t_i ($1 \le a_i \le n$; $1 \le t_i \le 10^4$). The problems are numbered from 1 to n.

Output

In the first line, output a single integer s — your maximum possible final score.

In the second line, output a single integer k ($0 \le k \le n$) — the number of problems you should solve.

In the third line, output k distinct integers $p_1, p_2, ..., p_k$ ($1 \le p_i \le n$) — the indexes of problems you should solve, in any order.

If there are several optimal sets of problems, you may output any of them.

input
```
5 300
3 100
4 150
4 80
2 90
2 300
```
output
```
2
3
3 1 4
```

input
```
2 100
1 787
2 788
```
output
```
0
0
```

input
```
2 100
2 42
2 58
```
output
```
2
2
1 2
```

In the first example, you should solve problems 3, 1, and 4. In this case you'll spend $80 + 100 + 90 = 270$ milliseconds, falling within the length of the exam, 300 milliseconds (and even leaving yourself 30 milliseconds to have a rest). Problems 3 and 1 will bring you a point each, while problem 4 won't. You'll score two points.

In the second example, the length of the exam is catastrophically not enough to solve even a single problem.

In the third example, you have just enough time to solve both problems in $42 + 58 = 100$ milliseconds and hand your solutions to the teacher with a smile.

G. Gennady the Dentist

1 second, 256 megabytes

Gennady is one of the best child dentists in Berland. Today n children got an appointment with him, they lined up in front of his office.

All children love to cry loudly at the reception at the dentist. We enumerate the children with integers from 1 to n in the order they go in the line. Every child is associated with the value of his *cofidence* p_i. The children take turns one after another to come into the office; each time the child that is the first in the line goes to the doctor.

While Gennady treats the teeth of the i-th child, the child is crying with the volume of v_i. At that the *confidence* of the first child in the line is reduced by the amount of v_i, the second one — by value $v_i - 1$, and so on. The children in the queue after the v_i-th child almost do not hear the crying, so their *confidence* remains unchanged.

If at any point in time the *confidence* of the j-th child is less than zero, he begins to cry with the volume of d_j and leaves the line, running towards the exit, without going to the doctor's office. At this the *confidence* of all the children after the j-th one in the line is reduced by the amount of d_j.

All these events occur immediately one after the other in some order. Some cries may lead to other cries, causing a chain reaction. Once in the hallway it is quiet, the child, who is first in the line, goes into the doctor's office.

Help Gennady the Dentist to determine the numbers of kids, whose teeth he will cure. Print their numbers in the chronological order.

Input

The first line of the input contains a positive integer n ($1 \le n \le 4000$) — the number of kids in the line.

Next n lines contain three integers each v_i, d_i, p_i ($1 \le v_i, d_i, p_i \le 10^6$) — the volume of the cry in the doctor's office, the volume of the cry in the hall and the *confidence* of the i-th child.

Output

In the first line print number k — the number of children whose teeth Gennady will cure.

In the second line print k integers — the numbers of the children who will make it to the end of the line in the increasing order.

input

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

output

```
2
1 3
```

input

```
5
4 5 1
5 3 9
4 1 2
2 1 8
4 1 9
```

output

```
4
1 2 4 5
```

In the first example, Gennady first treats the teeth of the first child who will cry with volume 4. The confidences of the remaining children will get equal to $-2, 1, 3, 1$, respectively. Thus, the second child also cries at the volume of 1 and run to the exit. The confidence of the remaining children will be equal to $0, 2, 0$. Then the third child will go to the office, and cry with volume 5. The other children won't bear this, and with a loud cry they will run to the exit.

In the second sample, first the first child goes into the office, he will cry with volume 4. The confidence of the remaining children will be equal to $5, -1, 6, 8$. Thus, the third child will cry with the volume of 1 and run to the exit. The confidence of the remaining children will be equal to $5, 5, 7$. After that, the second child goes to the office and cry with the volume of 5. The confidences of the remaining children will be equal to $0, 3$. Then the fourth child will go into the office and cry with the volume of 2. Because of this the confidence of the fifth child will be 1, and he will go into the office last.

H. Clique Problem

2 seconds, 256 megabytes

The clique problem is one of the most well-known NP-complete problems. Under some simplification it can be formulated as follows. Consider an undirected graph G. It is required to find a subset of vertices C of the maximum size such that any two of them are connected by an edge in graph G. Sounds simple, doesn't it? Nobody yet knows an algorithm that finds a solution to this problem in polynomial time of the size of the graph. However, as with many other NP-complete problems, the clique problem is easier if you consider a specific type of a graph.

Consider n distinct points on a line. Let the i-th point have the coordinate x_i and weight w_i. Let's form graph G, whose vertices are these points and edges connect exactly the pairs of points (i, j), such that the distance between them is not less than the sum of their weights, or more formally: $|x_i - x_j| \geq w_i + w_j$.

Find the size of the maximum clique in such graph.

Input

The first line contains the integer n $(1 \leq n \leq 200\,000)$ — the number of points.

Each of the next n lines contains two numbers x_i, w_i $(0 \leq x_i \leq 10^9, 1 \leq w_i \leq 10^9)$ — the coordinate and the weight of a point. All x_i are different.

Output

Print a single number — the number of vertexes in the maximum clique of the given graph.

input

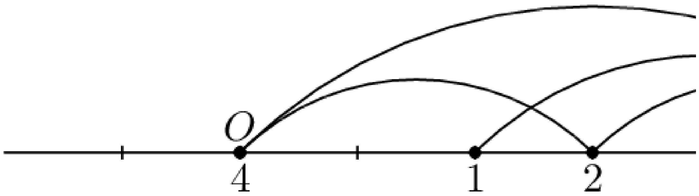
```
4
2 3
3 1
6 1
0 2
```

output

```
3
```

If you happen to know how to solve this problem without using the specific properties of the graph formulated in the problem statement, then you are able to get a prize of one million dollars!

The picture for the sample test.



I. Ralph And His Magic Field

1 second, 256 megabytes

Ralph has a magic field which is divided into $n \times m$ blocks. That is to say, there are n rows and m columns on the field. Ralph can put an integer in each block. However, the magic field doesn't always work properly. It works only if the product of integers in each row and each column equals to k, where k is either `1` or `-1`.

Now Ralph wants you to figure out the number of ways to put numbers in each block in such a way that the magic field works properly. Two ways are considered different if and only if there exists at least one block where the numbers in the first way and in the second way are different. You are asked to output the answer modulo $1000000007 = 10^9 + 7$.

Note that there is no range of the numbers to put in the blocks, but we can prove that the answer is not infinity.

Input

The only line contains three integers n, m and k $(1 \leq n, m \leq 10^{18}$, k is either `1` or `-1`).

Output

Print a single number denoting the answer modulo 1000000007.

input

```
1 1 -1
```

output

```
1
```

input

```
1 3 1
```

output

```
1
```

input

```
3 3 -1
```

output

```
16
```

In the first example the only way is to put `-1` into the only block.

In the second example the only way is to put `1` into every block.

J. Vasya And The Matrix

2 seconds, 256 megabytes

Now Vasya is taking an exam in mathematics. In order to get a good mark, Vasya needs to guess the matrix that the teacher has constructed!

Vasya knows that the matrix consists of n rows and m columns. For each row, he knows the xor (bitwise excluding or) of the elements in this row. The sequence $a_1, a_2, ..., a_n$ denotes the xor of elements in rows with indices $1, 2, ..., n$, respectively. Similarly, for each column, he knows the xor of the elements in this column. The sequence $b_1, b_2, ..., b_m$ denotes the xor of elements in columns with indices $1, 2, ..., m$, respectively.

Help Vasya! Find a matrix satisfying the given constraints or tell him that there is no suitable matrix.

Input

The first line contains two numbers n and m $(2 \le n, m \le 100)$ — the dimensions of the matrix.

The second line contains n numbers $a_1, a_2, ..., a_n$ $(0 \le a_i \le 10^9)$, where a_i is the xor of all elements in row i.

The third line contains m numbers $b_1, b_2, ..., b_m$ $(0 \le b_i \le 10^9)$, where b_i is the xor of all elements in column i.

Output

If there is no matrix satisfying the given constraints in the first line, output "NO".

Otherwise, on the first line output "YES", and then n rows of m numbers in each $c_{i1}, c_{i2}, \dots, c_{im}$ $(0 \le c_{ij} \le 2 \cdot 10^9)$ — the description of the matrix.

If there are several suitable matrices, it is allowed to print any of them.

input

```
2 3
2 9
5 3 13
```

output

```
YES
3 4 5
6 7 8
```

input

```
3 3
1 7 6
2 15 12
```

output

```
NO
```