A. Cards

2 seconds, 256 megabytes

When Serezha was three years old, he was given a set of cards with letters for his birthday. They were arranged into words in the way which formed the boy's mother favorite number in binary notation. Serezha started playing with them immediately and shuffled them because he wasn't yet able to read. His father decided to rearrange them. Help him restore the original number, on condition that it was the maximum possible one.

Input

The first line contains a single integer n ($1 \leqslant n \leqslant 10^5$) — the length of the string. The second line contains a string consisting of English lowercase letters: 'z', 'e', 'r', 'o' and 'n'.

It is guaranteed that it is possible to rearrange the letters in such a way that they form a sequence of words, each being either "`zero`" which corresponds to the digit 0 or "`one`" which corresponds to the digit 1.

Output

Print the maximum possible number in binary notation. Print binary digits separated by a space. The leading zeroes are allowed.

input
```
4
ezor
```
output
```
0
```

input
```
10
nznooeeoer
```
output
```
1 1 0
```

In the first example, the correct initial ordering is "`zero`".

In the second example, the correct initial ordering is "`oneonezero`".

B. Multiplication Table

2 seconds, 256 megabytes

Sasha grew up and went to first grade. To celebrate this event her mother bought her a multiplication table M with n rows and n columns such that $M_{ij} = a_i \cdot a_j$ where $a_1, \dots, a_n$ is some sequence of positive integers.

Of course, the girl decided to take it to school with her. But while she was having lunch, hooligan Grisha erased numbers on the main diagonal and threw away the array $a_1, \dots, a_n$. Help Sasha restore the array!

Input

The first line contains a single integer n ($3 \leqslant n \leqslant 10^3$), the size of the table.

The next n lines contain n integers each. The j-th number of the i-th line contains the number M_{ij} ($1 \leq M_{ij} \leq 10^9$). The table has zeroes on the main diagonal, that is, $M_{ii} = 0$.

Output

In a single line print n integers, the original array $a_1, \dots, a_n$ ($1 \leq a_i \leq 10^9$). It is guaranteed that an answer exists. If there are multiple answers, print any.

input
```
5
0 4 6 2 4
4 0 6 2 4
6 6 0 3 6
2 2 3 0 2
4 4 6 2 0
```
output
```
2 2 3 1 2
```

input
```
3
0 99990000 99970002
99990000 0 99980000
99970002 99980000 0
```
output
```
9999 10000 9998
```

C. Substring Game in the Lesson

2 seconds, 256 mebibytes

Mike and Ann are sitting in the classroom. The lesson is boring, so they decided to play an interesting game. Fortunately, all they need to play this game is a string s and a number k ($0 \leq k < |s|$).

At the beginning of the game, players are given a substring of s with left border l and right border r, both equal to k (i.e. initially $l = r = k$). Then players start to make moves one by one, according to the following rules:

- A player chooses l' and r' so that $l' \leq l$, $r' \geq r$ and $s[l', r']$ is lexicographically less than $s[l, r]$. Then the player changes l and r in this way: $l := l'$, $r := r'$.
- Ann moves first.
- The player, that can't make a move loses.

Recall that a substring $s[l, r]$ ($l \leq r$) of a string s is a continuous segment of letters from s that starts at position l and ends at position r. For example, "`ehn`" is a substring ($s[3, 5]$) of "`aaaehnsvz`" and "`ahz`" is not.

Mike and Ann were playing so enthusiastically that they did not notice the teacher approached them. Surprisingly, the teacher didn't scold them, instead of that he said, that he can figure out the winner of the game before it starts, even if he knows only s and k.

Unfortunately, Mike and Ann are not so keen in the game theory, so they ask you to write a program, that takes s and determines the winner for all possible k.

Input

The first line of the input contains a single string s ($1 \leq |s| \leq 5 \cdot 10^5$) consisting of lowercase English letters.

Output

Print $|s|$ lines.

In the line i write the name of the winner (print `Mike` or `Ann`) in the game with string s and $k = i$, if both play optimally

input
```
abba
```
output
```
Mike
Ann
Ann
Mike
```

input
```
cba
```

output

```
Mike
Mike
Mike
```

D. Alex and Julian

2 seconds, 256 megabytes

Boy Dima gave Julian a birthday present - set B consisting of positive integers. However, he didn't know, that Julian hates sets, but enjoys bipartite graphs more than anything else!

Julian was almost upset, but her friend Alex said, that he can build an undirected graph using this set in such way: let all integer numbers be vertices, then connect any two i and j with an edge if $|i-j|$ belongs to B.

Unfortunately, Julian doesn't like the graph, that was built using B. Alex decided to rectify the situation, so he wants to erase some numbers form B, so that graph built using the new set is bipartite. The difficulty of this task is that the graph, Alex has to work with, has an infinite number of vertices and edges! It is impossible to solve this task alone, so Alex asks you for help. Write a program that erases a subset of **minimum** size from B so that graph constructed on the new set is bipartite.

Recall, that graph is bipartite if all its vertices can be divided into two disjoint sets such that every edge connects a vertex from different sets.

Input

First line contains an integer n $(1 \leqslant n \leqslant 200\,000)$ — size of B

Second line contains n integers $b_1, b_2, \ldots, b_n$ $(1 \leqslant b_i \leqslant 10^{18})$ — numbers of B, all b_i are **unique**

Output

In first line print single integer k – number of erased elements. In second line print k integers – values of erased elements.

If there are multiple answers, print any of them.

input

```
3
1 2 3
```

output

```
1
2
```

input

```
2
2 6
```

output

```
0
```

E. Tourism

2 seconds, 256 megabytes

Alex decided to go on a touristic trip over the country.

For simplicity let's assume that the country has n cities and m bidirectional roads connecting them. Alex lives in city s and initially located in it. To compare different cities Alex assigned each city a score w_i which is as high as interesting city seems to Alex.

Alex believes that his trip will be interesting only if he will not use any road twice in a row. That is if Alex came to city v from city u, he may choose as the next city in the trip any city connected with v by the road, except for the city u.

Your task is to help Alex plan his city in a way that maximizes total score over all cities he visited. Note that for each city its score is counted at most once, even if Alex been there several times during his trip.

Input

First line of input contains two integers n and m, $(1 \le n \le 2 \cdot 10^5$, $0 \le m \le 2 \cdot 10^5)$ which are numbers of cities and roads in the country.

Second line contains n integers $w_1, w_2, \ldots, w_n$ $(0 \le w_i \le 10^9)$ which are scores of all cities.

The following m lines contain description of the roads. Each of these m lines contains two integers u and v $(1 \le u, v \le n)$ which are cities connected by this road.

It is guaranteed that there is at most one direct road between any two cities, no city is connected to itself by the road and, finally, it is possible to go from any city to any other one using only roads.

The last line contains single integer s $(1 \le s \le n)$, which is the number of the initial city.

Output

Output single integer which is the maximum possible sum of scores of visited cities.

input

```
5 7
2 2 8 6 9
1 2
1 3
2 4
3 2
4 5
2 5
1 5
2
```

output

```
27
```

input

```
10 12
1 7 1 9 3 3 6 30 1 10
1 2
1 3
3 5
5 7
2 3
5 4
6 9
4 6
3 7
6 8
9 4
9 10
6
```

output

```
61
```

F. Gardener Alex

2 seconds, 256 megabytes

Gardener Alex loves to grow trees. We remind that tree is a connected acyclic graph on n vertices.

Today he decided to grow a rooted binary tree. A binary tree is a tree where any vertex has no more than two sons. Luckily, Alex has a permutation of numbers from 1 to n which he was presented at his last birthday, so he decided to grow a tree according to this permutation. To do so he does the following process: he finds a minimum element and makes it a root of the tree. After that permutation is divided into two parts: everything that is to the left of the minimum element, and everything that is to the right. The minimum element on the left part becomes the left son of the root, and the minimum element on the right part becomes the right son of the root. After that, this process is repeated recursively on both parts.

Now Alex wants to grow a forest of trees: one tree for each cyclic shift of the permutation. He is interested in what cyclic shift gives the tree of minimum depth. Unfortunately, growing a forest is a hard and long process, but Alex wants the answer right now. Will you help him?

We remind that cyclic shift of permutation $a_1, a_2, \ldots, a_k, \ldots, a_n$ for k elements to the left is the permutation $a_{k+1}, a_{k+2}, \ldots, a_n, a_1, a_2, \ldots, a_k$.

Input

First line contains an integer number n $(1 \leqslant n \leqslant 200\,000)$ — length of the permutation.

Second line contains n integer numbers $a_1, a_2, \ldots, a_n$ $(1 \leqslant a_i \leqslant n)$, and it is guaranteed that all numbers occur exactly one time.

Output

Print two numbers separated with space: minimum possible depth of a tree and how many elements we need to shift left to achieve this depth. The number of elements should be a number from 0 to $n-1$. If there are several possible answers, print any of them.

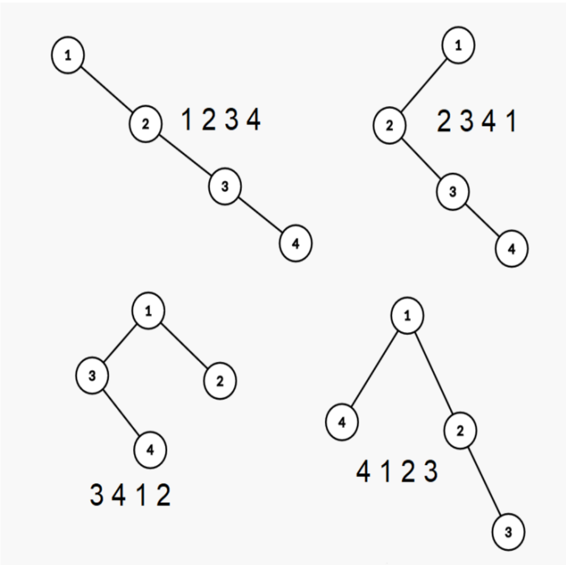
input

```
4
1 2 3 4
```

output

```
3 2
```

The following picture depicts all possible trees for sample test and cyclic shifts on which they are achieved.



G. Geolocation

2 seconds, 256 megabytes

You are working for the Gryzzl company, headquartered in Pawnee, Indiana.

The new national park has been opened near Pawnee recently and you are to implement a geolocation system, so people won't get lost. The concept you developed is innovative and minimalistic. There will be n antennas located somewhere in the park. When someone would like to know their current location, their Gryzzl hologram phone will communicate with antennas and obtain distances from a user's current location to all antennas.

Knowing those distances and antennas locations it should be easy to recover a user's location... Right? Well, almost. The only issue is that there is no way to distinguish antennas, so you don't know, which distance corresponds to each antenna. Your task is to find a user's location given as little as all antennas location and an unordered multiset of distances.

Input

The first line of input contains a single integer n ($2 \le n \le 10^5$) which is the number of antennas.

The following n lines contain coordinates of antennas, i-th line contain two integers x_i and y_i ($0 \le x_i, y_i \le 10^8$). It is guaranteed that no two antennas coincide.

The next line of input contains integer m ($1 \le n \cdot m \le 10^5$), which is the number of queries to determine the location of the user.

Following m lines contain n integers $0 \le d_1 \le d_2 \le \cdots \le d_n \le 2 \cdot 10^{16}$ each. These integers form a multiset of **squared** distances from unknown user's location $(x;y)$ to antennas.

For all test cases except the examples it is guaranteed that all user's locations $(x;y)$ were chosen uniformly at random, independently from each other among all possible integer locations having $0 \le x, y \le 10^8$.

Output

For each query output k, the number of possible a user's locations matching the given input and then output the list of these locations in lexicographic order.

It is guaranteed that the sum of all k over all points does not exceed 10^6.

input

```
3
0 0
0 1
1 0
1
1 1 2
```

output

```
1 1 1
```

input

```
4
0 0
0 1
1 0
1 1
2
0 1 1 2
2 5 5 8
```

output

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

As you see in the second example, although initially a user's location is picked to have non-negative coordinates, you have to output all possible integer locations.