A. Zebras

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Oleg writes down the history of the days he lived. For each day he decides if it was good or bad. Oleg calls a non-empty sequence of days a *zebra*, if it starts with a bad day, ends with a bad day, and good and bad days are alternating in it. Let us denote bad days as `0` and good days as `1`. Then, for example, sequences of days `0`, `010`, `01010` are zebras, while sequences `1`, `0110`, `0101` are not.

Oleg tells you the story of days he lived in chronological order in form of string consisting of `0` and `1`. Now you are interested if it is possible to divide Oleg's life history into several **subsequences**, each of which is a zebra, and the way it can be done. Each day must belong to exactly one of the subsequences. For each of the subsequences, days forming it must be ordered chronologically. Note that subsequence does not have to be a group of consecutive days.

Input

In the only line of input data there is a non-empty string s consisting of characters `0` and `1`, which describes the history of Oleg's life. Its length (denoted as $|s|$) does not exceed $200\ 000$ characters.

Output

If there is a way to divide history into zebra subsequences, in the first line of output you should print an integer k ($1 \le k \le |s|$), the resulting number of subsequences. In the i-th of following k lines first print the integer l_i ($1 \le l_i \le |s|$), which is the length of the i-th subsequence, and then l_i indices of days forming the subsequence. Indices must follow in ascending order. Days are numbered starting from 1. Each index from 1 to n must belong to exactly one subsequence. If there is no way to divide day history into zebra subsequences, print `-1`.

Subsequences may be printed in any order. If there are several solutions, you may print any of them. You do not have to minimize nor maximize the value of k.

Examples

input

```
0010100
```

output

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

input

```
111
```

output

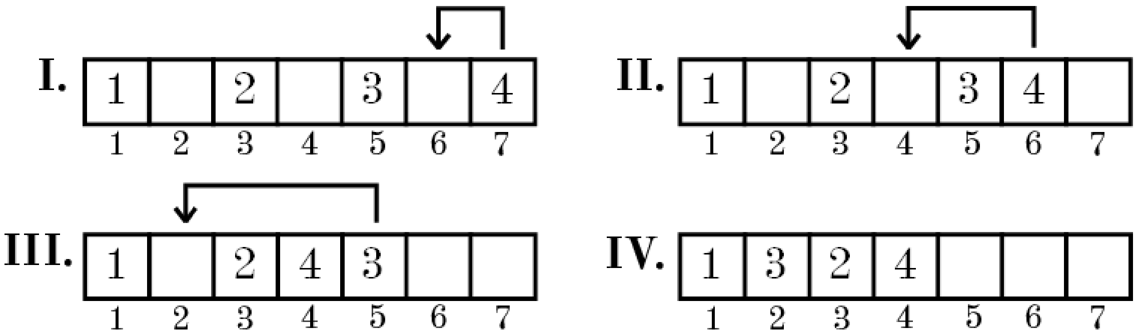
```
-1
```

B. A Leapfrog in the Array

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Dima is a beginner programmer. During his working process, he regularly has to repeat the following operation again and again: to remove every second element from the array. One day he has been bored with easy solutions of this problem, and he has come up with the following extravagant algorithm.

Let's consider that initially array contains n numbers from 1 to n and the number i is located in the cell with the index $2i - 1$ (Indices are numbered starting from one) and other cells of the array are empty. Each step Dima selects a non-empty array cell with the maximum index and moves the number written in it to the nearest empty cell to the left of the selected one. The process continues until all n numbers will appear in the first n cells of the array. For example if $n = 4$, the array is changing as follows:



You have to write a program that allows you to determine what number will be in the cell with index x ($1 \le x \le n$) after Dima's algorithm finishes.

Input

The first line contains two integers n and q ($1 \le n \le 10^{18}$, $1 \le q \le 200\,000$), the number of elements in the array and the number of queries for which it is needed to find the answer.

Next q lines contain integers x_i ($1 \le x_i \le n$), the indices of cells for which it is necessary to output their content after Dima's algorithm finishes.

Output

For each of q queries output one integer number, the value that will appear in the corresponding array cell after Dima's algorithm finishes.

Examples

input

```
4 3
2
3
4
```

output

```
3
2
4
```

input

```
13 4
10
5
4
8
```

output

```
13
3
8
9
```

Note

The first example is shown in the picture.

In the second example the final array is $[1, 12, 2, 8, 3, 11, 4, 9, 5, 13, 6, 10, 7]$.

C. Data Center Maintenance

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

BigData Inc. is a corporation that has n data centers indexed from 1 to n that are located all over the world. These data centers provide storage for client data (you can figure out that client data is really big!).

Main feature of services offered by BigData Inc. is the access availability guarantee even under the circumstances of any data center having an outage. Such a guarantee is ensured by using the *two-way replication*. Two-way replication is such an approach for data storage that any piece of data is represented by two identical copies that are stored in two different data centers.

For each of m company clients, let us denote indices of two different data centers storing this client data as $c_{i,1}$ and $c_{i,2}$.

In order to keep data centers operational and safe, the software running on data center computers is being updated regularly. Release cycle of BigData Inc. is one day meaning that the new version of software is being deployed to the data center computers each day.

Data center software update is a non-trivial long process, that is why there is a special hour-long time frame that is dedicated for data center maintenance. During the maintenance period, data center computers are installing software updates, and thus they may be unavailable. Consider the day to be exactly h hours long. For each data center there is an integer u_j ($0 \le u_j \le h - 1$) defining the index of an hour of day, such that during this hour data center j is unavailable due to maintenance.

Summing up everything above, the condition $u_{c_{i,1}} \neq u_{c_{i,2}}$ should hold for each client, or otherwise his data may be unaccessible while data centers that store it are under maintenance.

Due to occasional timezone change in different cities all over the world, the maintenance time in some of the data centers may change by one hour sometimes. Company should be prepared for such situation, that is why they decided to conduct an experiment, choosing some non-empty subset of data centers, and shifting the maintenance time for them by an hour later (i.e. if $u_j = h - 1$, then the new maintenance hour would become 0, otherwise it would become $u_j + 1$). Nonetheless, such an experiment should not break the accessibility guarantees, meaning that data of any client should be still available during any hour of a day after the data center maintenance times are changed.

Such an experiment would provide useful insights, but changing update time is quite an expensive procedure, that is why the company asked you to find out the minimum number of data centers that have to be included in an experiment in order to keep the data accessibility guarantees.

Input

The first line of input contains three integers n, m and h ($2 \le n \le 100\,000$, $1 \le m \le 100\,000$, $2 \le h \le 100\,000$), the number of company data centers, number of clients and the day length of day measured in hours.

The second line of input contains n integers $u_1, u_2, \ldots, u_n$ ($0 \le u_j < h$), j-th of these numbers is an index of a maintenance hour for data center j.

Each of the next m lines contains two integers $c_{i,1}$ and $c_{i,2}$ ($1 \le c_{i,1}, c_{i,2} \le n$, $c_{i,1} \neq c_{i,2}$), defining the data center indices containing the data of client i.

It is guaranteed that the given maintenance schedule allows each client to access at least one copy of his data at any moment of day.

Output

In the first line print the minimum possible number of data centers k ($1 \le k \le n$) that have to be included in an experiment in order to keep the data available for any client.

In the second line print k distinct integers $x_1, x_2, \ldots, x_k$ ($1 \le x_i \le n$), the indices of data centers whose maintenance time will be shifted by one hour later. Data center indices may be printed in any order.

If there are several possible answers, it is allowed to print any of them. It is guaranteed that at there is at least one valid choice of data centers.

Examples

input

```
3 3 5
4 4 0
1 3
3 2
3 1
```

output

```
1
3
```

input

```
4 5 4
2 1 0 3
4 3
3 2
1 2
1 4
1 3
```

output

```
4
1 2 3 4
```

Note

Consider the first sample test. The given answer is the only way to conduct an experiment involving the only data center. In such a scenario the third data center has a maintenance during the hour 1, and no two data centers storing the information of the same client have maintenance at the same hour.

On the other hand, for example, if we shift the maintenance time on hour later for the first data center, then the data of clients 1 and 3 will be unavailable during the hour 0.

D. Curfew

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Instructors of Some Informatics School make students go to bed.

The house contains n rooms, in each room exactly b students were supposed to sleep. However, at the time of curfew it happened that many students are not located in their assigned rooms. The rooms are arranged in a row and numbered from 1 to n. Initially, in i-th room there are a_i

students. All students are currently somewhere in the house, therefore $a_1 + a_2 + \ldots + a_n = nb$. Also 2 instructors live in this house.

The process of curfew enforcement is the following. One instructor starts near room 1 and moves toward room n, while the second instructor starts near room n and moves toward room 1. After processing current room, each instructor moves on to the next one. Both instructors enter rooms and move simultaneously, if n is odd, then only the first instructor processes the middle room. When all rooms are processed, the process ends.

When an instructor processes a room, she counts the number of students in the room, then turns off the light, and locks the room. Also, if the number of students inside the processed room is not equal to b, the instructor writes down the number of this room into her notebook (and turns off the light, and locks the room). Instructors are in a hurry (to prepare the study plan for the next day), so they don't care about who is in the room, but only about the number of students.

While instructors are inside the rooms, students can run between rooms that are not locked and not being processed. A student can run by at most d rooms, that is she can move to a room with number that differs my at most d. Also, after (or instead of) running each student can hide under a bed in a room she is in. In this case the instructor will not count her during the processing. In each room any number of students can hide simultaneously.

Formally, here is what's happening:

- A curfew is announced, at this point in room i there are a_i students.
- Each student can run to another room but not further than d rooms away from her initial room, or stay in place. After that each student can optionally hide under a bed.
- Instructors enter room 1 and room n, they count students there and lock the room (after it no one can enter or leave this room).
- Each student from rooms with numbers from 2 to $n - 1$ can run to another room but not further than d rooms away from her **current** room, or stay in place. Each student can optionally hide under a bed.
- Instructors move from room 1 to room 2 and from room n to room $n - 1$.
- This process continues until all rooms are processed.

Let x_1 denote the number of rooms in which the first instructor counted the number of non-hidden students different from b, and x_2 be the same number for the second instructor. Students know that the principal will only listen to one complaint, therefore they want to minimize the maximum of numbers x_i. Help them find this value if they use the optimal strategy.

Input

The first line contains three integers n, d and b ($2 \le n \le 100\,000$, $1 \le d \le n - 1$, $1 \le b \le 10\,000$), number of rooms in the house, running distance of a student, official number of students in a room.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$), i-th of which stands for the number of students in the i-th room before curfew announcement.

It is guaranteed that $a_1 + a_2 + \ldots + a_n = nb$.

Output

Output one integer, the minimal possible value of the maximum of x_i.

Examples

input

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

output

```
1
```

input

```
6 1 2
3 8 0 1 0 0
```

output

```
2
```

Note

In the first sample the first three rooms are processed by the first instructor, and the last two are processed by the second instructor. One of the optimal strategies is the following: firstly three students run from room 5 to room 4, on the next stage two of them run to room 3, and one of those two hides under a bed. This way, the first instructor writes down room 2, and the second writes down nothing.

In the second sample one of the optimal strategies is the following: firstly all students in room 1 hide, all students from room 2 run to room 3. On the next stage one student runs from room 3 to room 4, and 5 students hide. This way, the first instructor writes down rooms 1 and 2, the second instructor writes down rooms 5 and 6.

E. Binary Cards

time limit per test: 1 second

memory limit per test: 512 megabytes

input: standard input

output: standard output

It is never too late to play the fancy "Binary Cards" game!

There is an infinite amount of cards of positive and negative ranks that are used in the game. The absolute value of any card rank is a power of two, i.e. each card has a rank of either 2^k or -2^k for some integer $k \ge 0$. There is an infinite amount of cards of any valid rank.

At the beginning of the game player forms his deck that is some multiset (possibly empty) of cards. It is allowed to pick any number of cards of any rank but the small deck is considered to be a skill indicator. Game consists of n rounds. In the i-th round jury tells the player an integer a_i. After that the player is obligated to draw such a subset of his deck that the sum of ranks of the chosen cards is equal to a_i (it is allowed to not draw any cards, in which case the sum is considered to be equal to zero). If player fails to do so, he loses and the game is over. Otherwise, player takes back all of his cards into his deck and the game proceeds to the next round. Player is considered a winner if he is able to draw the suitable set of cards in each of the rounds.

Somebody told you which numbers a_i the jury is going to tell you in each round. Now you want to pick a deck consisting of the minimum number of cards that allows you to win the "Binary Cards" game.

Input

The first line of input contains an integer n ($1 \le n \le 100\,000$), the number of rounds in the game.

The second line of input contains n integers $a_1, a_2, \ldots, a_n$ ($-100\,000 \le a_i \le 100\,000$), the numbers that jury is going to tell in each round.

Output

In the first line print the integer k ($0 \le k \le 100\,000$), the minimum number of cards you have to pick in your deck in ordered to win the "Binary Cards".

In the second line print k integers $b_1, b_2, \ldots, b_k$ ($-2^{20} \le b_i \le 2^{20}$, $|b_i|$ is a power of two), the ranks of the cards in your deck. You may output ranks in any order. If there are several optimum decks, you are allowed to print any of them.

It is guaranteed that there exists a deck of minimum size satisfying all the requirements above.

Examples

input

```
1
9
```

output

```
2
1 8
```

input

```
5
-1 3 0 4 7
```

output

```
3
4 -1 4
```

input

```
4
2 -2 14 18
```

output

```
3
-2 2 16
```

Note

In the first sample there is the only round in the game, in which you may simply draw both your cards. Note that this sample test is the only one satisfying the first test group constraints.

In the second sample you may draw the only card -1 in the first round, cards 4 and -1 in the second round, nothing in the third round, the only card 4 in the fourth round and the whole deck in the fifth round.

F. Astronomy

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Year 18 AD. Famous astronomer Philon the Berlander publishes a book "About Sky and Cosmos», in which he describes an incredible picture seen by him on a night sky while observing the skies. He once seen $2n$ stars on a clear sky and the Moon. Surprisingly, it was possible to divide all stars in pairs in such way that any line passing through the centers of two paired stars also passed through the center of the Moon, also all such lines were distinct. Philon carefully represented such a situation on a sky map, introducing a coordinate system. While doing that, he noticed that centers of all stars and the center of the Moon are points with integer coordinates. As Philon thought that the Earth and the Moon were flat, his coordinate system was two-dimensional. Coordinate system was chosen by an astronomer in such way that the coordinates of all objects (including the Moon) are no more than 10^6 by the absolute value. Moreover, no two objects (two stars or a star and a Moon) were not located at the same point.

Apart from the sky map Philon wrote a prediction that in 2000 years stars will take exactly the same places, but the Moon will be replaced by a huge comet which will destroy the Earth.

It is 2018 AD now. You got a book of Philon the Berlander and you were horrified to discover that the stars on the sky are in exactly the same position as they were 2000 years ago! Unfortunately, some parts of a sky map were lost, so there are only star locations that are visible on it and there are no details about how the stars were divided in pairs. Moreover,, there is no point corresponding to the center of the Moon on this map. In order to find out the possible location of a comet and save the humanity from the terrible end, you should immediately find out some suitable location for the center of the Moon!

Input

In the first line of input there is an integer n ($2 \le n \le 2600$), the number of star pairs seen by an astronomer in the sky.

In the following $2n$ lines there are pairs of integers x_i, y_i ($-10^6 \le x_i, y_i \le 10^6$), the coordinates of star centers in the sky map. Note that the stars are listed in an arbitrary order that has nothing common with the distribution of stars into pairs as found out by Philon the Berlander. Centers of no two stars coincide.

Output

If astronomer was wrong and there is no way to distribute all points in pairs in such manner that all lines passing through these point pairs are distinct lines intersecting in a single point with integer coordinates, and this intersection point is different from centers of all stars, then print "`No`" (without the quotes) in the only line of input.

Otherwise print "`Yes`" (without the quotes). In the second line print pair of integers x, y ($|x|, |y| \le 10^6$), the coordinates of a point that contains a center of the Moon in your solution. If there are several possible answers, output any of them. Note that the printed point should be different from all the star centers.

Examples

input
```
2
1 1
1 3
3 1
3 3
```
output
```
Yes
2 2
```

input
```
3
4 2
6 2
2 1
2 6
4 4
6 6
```
output
```
Yes
2 2
```

input
```
2
1 1
2 2
4 4
5 5
```
output
```
No
```

input
```
2
1 1
2 1
1 2
2 2
```
output
```
No
```

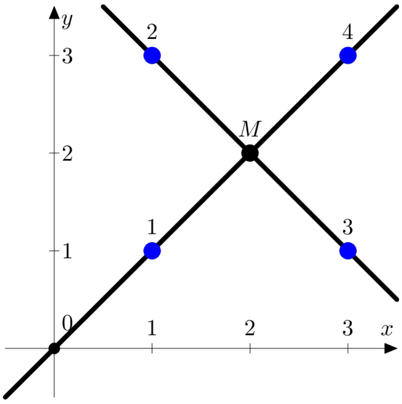
input
```
2
0 0
1 1
-1 -1
-1 1
```

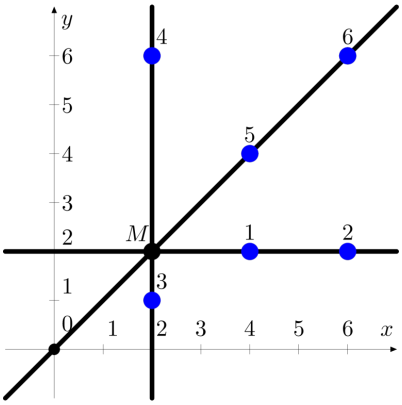
output

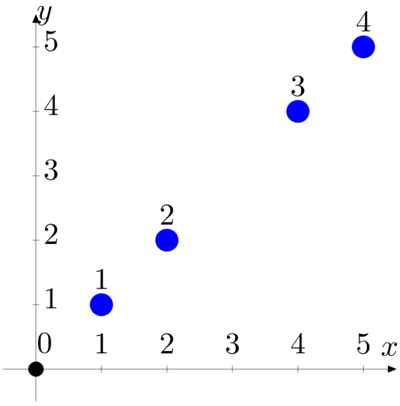
```
No
```

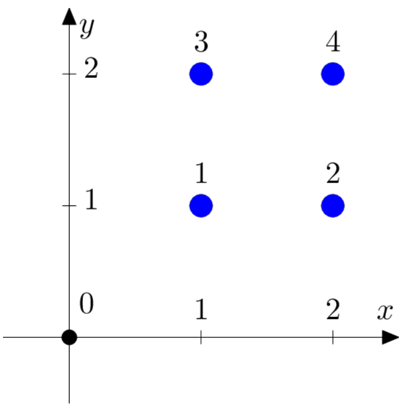
Note

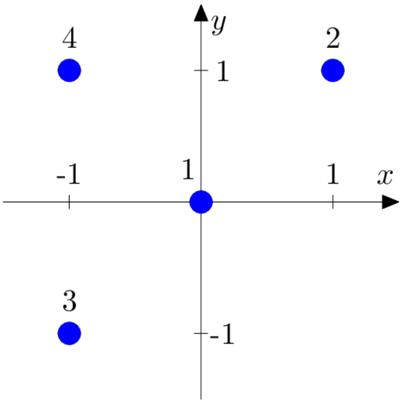
Pictures for the sample tests are given below.











In the fourth sample test the Moon center could not have possibly been in a point $(1.5,\ 1.5)$ since the coordinates of this point are non-integer.

In the fifth sample test there are no suitable points that do not coincide with a center of some star.