CODEFORCES β
Sponsored by Telegram

Codeforces Round #469 (Div. 2)

A. Left-handers, Right-handers and Ambidexters

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are at a water bowling training. There are l people who play with their left hand, r people, who play with their right hand, and a ambidexters, who can play with left or right hand.

The coach decided to form a team of even number of players, exactly half of the players should play with their right hand, and exactly half of the players should play with their left hand. One player should use only on of his hands.

Ambidexters play as well with their right hand as with their left hand. In the team, an ambidexter can play with their left hand, or with their right hand.

Please find the maximum possible size of the team, where equal number of players use their left and right hands, respectively.

Input
The only line contains three integers l, r and a ($0 \le l, r, a \le 100$) — the number of left-handers, the number of right-handers and the number of ambidexters at the training.

Output
Print a single even integer — the maximum number of players in the team. It is possible that the team can only have zero number of players.

Examples

input	Copy
1 4 2	
output	
6	

input	Copy
5 5 5	
output	
14	

input	Copy
0 2 0	
output	
0	

Note
In the first example you can form a team of 6 players. You should take the only left-hander and two ambidexters to play with left hand, and three right-handers to play with right hand. The only person left can't be taken into the team.

In the second example you can form a team of 14 people. You have to take all five left-handers, all five right-handers, two ambidexters to play with left hand and two ambidexters to play with right hand.

B. Intercepted Message

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

Hacker Zhorik wants to decipher two secret messages he intercepted yesterday. Yeah message is a sequence of encrypted blocks, each of them consists of several bytes of information.

Zhorik knows that each of the messages is an archive containing one or more files. Zhorik knows how each of these archives was transferred through the network: if an archive consists of k files of sizes $l_1, l_2, ..., l_k$ bytes, then the i-th file is split to one or more blocks $b_{i,1}, b_{i,2}, ..., b_{i,m_i}$ (here the total length of the blocks $b_{i,1} + b_{i,2} + ... + b_{i,m_i}$ is equal to the length of the file l_i), and after that all blocks are transferred through the network, maintaining the order of files in the archive.

Zhorik thinks that the two messages contain the same archive, because their total lengths are equal. However, each file can be split in blocks in different ways in the two messages.

You are given the lengths of blocks in each of the two messages. Help Zhorik to determine what is the maximum number of files could be in the archive, if the Zhorik's assumption is correct.

Input

The first line contains two integers n, m ($1 \le n, m \le 10^5$) — the number of blocks in the first and in the second messages.

The second line contains n integers $x_1, x_2, ..., x_n$ ($1 \le x_i \le 10^6$) — the length of the blocks that form the first message.

The third line contains m integers $y_1, y_2, ..., y_m$ ($1 \le y_i \le 10^6$) — the length of the blocks that form the second message.

It is guaranteed that $x_1 + ... + x_n = y_1 + ... + y_m$. **Also, it is guaranteed that $x_1 + ... + x_n \le 10^6$.**

Output

Print the maximum number of files the intercepted array could consist of.

Examples

input Copy

```
7 6
2 5 3 1 11 4 4
7 8 2 4 1 8
```

output

```
3
```

input Copy

```
3 3
1 10 100
1 100 10
```

output

```
2
```

input Copy

```
1 4
4
1 1 1 1
```

output

```
1
```

Note

In the first example the maximum number of files in the archive is 3. For example, it is possible that in the archive are three files of sizes $2 + 5 = 7$, $15 = 3 + 1 + 11 = 8 + 2 + 4 + 1$ and $4 + 4 = 8$.

In the second example it is possible that the archive contains two files of sizes 1 and $110 = 10 + 100 = 100 + 10$. Note that the order of files is kept while transferring archives through the network, so we can't say that there are three files of sizes 1, 10 and 100.

In the third example the only possibility is that the archive contains a single file of size 4.

C. 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	Copy
0010100	

output
3 3 1 3 4 3 2 5 6 1 7

input	Copy
111	

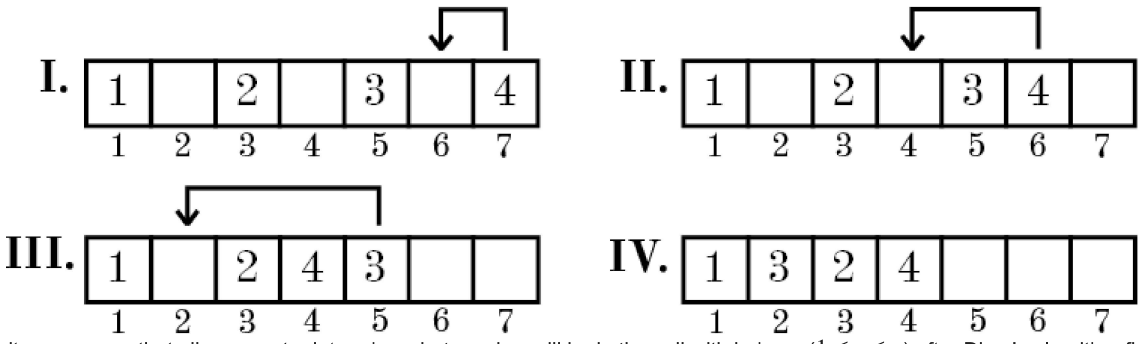
output
-1

D. 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	Copy
4 3 2 3 4	
output	
3 2 4	

input	Copy
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 $[1, 12, 2, 8, 3, 11, 4, 9, 5, 13, 6, 10, 7]$.

E. 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 Copy

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

output

```
1
3
```

input Copy

```
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.

F. 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 $a_1 + a_2 + ... + a_n = nb$. 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, ..., 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 $a_1 + a_2 + ... + a_n = nb$.

Output

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

Examples

input	Copy
5 1 1 1 0 0 0 4	

output
1

input	Copy
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.
