A. Snacktower

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

According to an old legeng, a long time ago Ankh-Morpork residents did something wrong to miss Fortune, and she cursed them. She said that at some time n snacks of distinct sizes will fall on the city, and the residents should build a Snacktower of them by placing snacks one on another. Of course, big snacks should be at the bottom of the tower, while small snacks should be at the top.

Years passed, and once different snacks started to fall onto the city, and the residents began to build the Snacktower.

However, they faced some troubles. Each day exactly one snack fell onto the city, but their order was strange. So, at some days the residents weren't able to put the new stack on the top of the Snacktower: they had to wait until all the bigger snacks fell. Of course, in order to not to anger miss Fortune again, the residents placed each snack on the top of the tower immediately as they could do it.

Write a program that models the behavior of Ankh-Morpork residents.

Input

The first line contains single integer n $(1 \le n \le 100\ 000)$ — the total number of snacks.

The second line contains n integers, the i-th of them equals the size of the snack which fell on the i-th day. Sizes are distinct integers from 1 to n.

Output

Print n lines. On the i-th of them print the sizes of the snacks which the residents placed on the top of the Snacktower on the i-th day in the order they will do that. If no snack is placed on some day, leave the corresponding line empty.

Examples

input

```
3
3 1 2
```

output

```
3

2 1
```

input

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

output

```

5 4


3 2 1
```

Note

In the example a snack of size 3 fell on the first day, and the residents immediately placed it. On the second day a snack of size 1 fell, and the residents weren't able to place it because they were missing the snack of size 2. On the third day a snack of size 2 fell, and the residents immediately placed it. Right after that they placed the snack of size 1 which had fallen before.

B. The Queue

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Finally! Vasya have come of age and that means he can finally get a passport! To do it, he needs to visit the passport office, but it's not that simple. There's only one receptionist at the passport office and people can queue up long before it actually opens. Vasya wants to visit the passport office tomorrow.

He knows that the receptionist starts working after t_s minutes have passed after midnight and closes after t_f minutes have passed after midnight (so that $(t_f - 1)$ is the last minute when the receptionist is still working). The receptionist spends exactly t minutes on each person in the queue. If the receptionist would stop working within t minutes, he stops serving visitors (other than the one he already serves).

Vasya also knows that exactly n visitors would come tomorrow. For each visitor Vasya knows the point of time when he would come to the passport office. Each visitor queues up and doesn't leave until he was served. If the receptionist is free when a visitor comes (in particular, if the previous visitor was just served and the queue is empty), the receptionist begins to serve the newcomer immediately.

"Reception 1"

For each visitor, the point of time when he would come to the passport office is positive. Vasya can come to the office at the time zero (that is, at midnight) if he needs so, but he can come to the office only at integer points of time. If Vasya arrives at the passport office at the same time with several other visitors, he yields to them and stand in the queue after the last of them.

Vasya wants to come at such point of time that he will be served by the receptionist, and he would spend the minimum possible time in the queue. Help him!

Input

The first line contains three integers: the point of time when the receptionist begins to work t_s, the point of time when the receptionist stops working t_f and the time the receptionist spends on each visitor t. The second line contains one integer n — the amount of visitors $(0 \le n \le 100\ 000)$. The third line contains positive integers in non-decreasing order — the points of time when the visitors arrive to the passport office.

All times are set in minutes and do not exceed 10^{12}; it is guaranteed that $t_s < t_f$. It is also guaranteed that Vasya can arrive at the passport office at such a point of time that he would be served by the receptionist.

Output

Print single non-negative integer — the point of time when Vasya should arrive at the passport office. If Vasya arrives at the passport office at the same time with several other visitors, he yields to them and queues up the last. If there are many answers, you can print any of them.

Examples

input

```
10 15 2
2
10 13
```

output

```
12
```

input

```
8 17 3
4
3 4 5 8
```

output

```
2
```

Note

In the first example the first visitor comes exactly at the point of time when the receptionist begins to work, and he is served for two minutes. At 12 minutes after the midnight the receptionist stops serving the first visitor, and if Vasya arrives at this moment, he will be served immediately, because the next visitor would only come at 13 minutes after midnight.

In the second example, Vasya has to come before anyone else to be served.

C. Garland

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Once at New Year Dima had a dream in which he was presented a fairy garland. A garland is a set of lamps, some pairs of which are connected by wires. Dima remembered that each two lamps in the garland were connected directly or indirectly via some wires. Furthermore, the number of wires was exactly one less than the number of lamps.

There was something unusual about the garland. Each lamp had its own brightness which depended on the temperature of the lamp. Temperatures could be positive, negative or zero. Dima has two friends, so he decided to share the garland with them. He wants to cut two different wires so that the garland breaks up into three parts. Each part of the garland should shine equally, i. e. the sums of lamps' temperatures should be equal in each of the parts. Of course, each of the parts should be non-empty, i. e. each part should contain at least one lamp.

Help Dima to find a suitable way to cut the garland, or determine that this is impossible.

While examining the garland, Dima lifted it up holding by one of the lamps. Thus, each of the lamps, except the one he is holding by, is now hanging on some wire. So, you should print two lamp ids as the answer which denote that Dima should cut the wires these lamps are hanging on. Of course, the lamp Dima is holding the garland by can't be included in the answer.

Input

The first line contains single integer n $(3 \le n \le 10^6)$ — the number of lamps in the garland.

Then n lines follow. The i-th of them contain the information about the i-th lamp: the number lamp a_i, it is hanging on (and 0, if is there is no such lamp), and its temperature t_i ($-100 \le t_i \le 100$). The lamps are numbered from 1 to n.

Output

If there is no solution, print `-1`.

Otherwise print two integers — the indexes of the lamps which mean Dima should cut the wires they are hanging on. If there are multiple answers, print any of them.

Examples

input

```
6
2 4
0 5
4 2
2 1
1 1
4 2
```

output

```
1 4
```

input

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

output

```
-1
```

Note

The garland and cuts scheme for the first example:

D. Cartons of milk

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

Olya likes milk very much. She drinks k cartons of milk each day if she has at least k and drinks all of them if she doesn't. But there's an issue — expiration dates. Each carton has a date after which you can't drink it (you still can drink it exactly at the date written on the carton). Due to this, if Olya's fridge contains a carton past its expiry date, she throws it away.

Olya hates throwing out cartons, so when she drinks a carton, she chooses the one which expires the fastest. It's easy to understand that this strategy minimizes the amount of cartons thrown out and lets her avoid it if it's even possible.

Milk. Best before: 20.02.2017.

The main issue Olya has is the one of buying new cartons. Currently, there are n cartons of milk in Olya's fridge, for each one an expiration date is known (how soon does it expire, measured in days). In the shop that Olya visited there are m cartons, and the expiration date is known for each of those cartons as well.

Find the maximum number of cartons Olya can buy so that she wouldn't have to throw away any cartons. Assume that Olya drank no cartons today.

Input

In the first line there are three integers n, m, k ($1 \le n, m \le 10^6$, $1 \le k \le n + m$) — the amount of cartons in Olya's fridge, the amount of cartons in the shop and the number of cartons Olya drinks each day.

In the second line there are n integers $f_1, f_2, ..., f_n$ ($0 \le f_i \le 10^7$) — expiration dates of the cartons in Olya's fridge. The expiration date is expressed by the number of days the drinking of this carton can be delayed. For example, a 0 expiration date means it must be drunk today, 1 — no later than tomorrow, etc.

In the third line there are m integers $s_1, s_2, ..., s_m$ ($0 \le s_i \le 10^7$) — expiration dates of the cartons in the shop in a similar format.

Output

If there's no way for Olya to drink the cartons she already has in her fridge, print `-1`.

Otherwise, in the first line print the maximum number x of cartons which Olya can buy so that she wouldn't have to throw a carton away. The next line should contain exactly x integers — the numbers of the cartons that should be bought (cartons are numbered in an order in which they are written in the input, starting with 1). Numbers should not repeat, but can be in arbitrary order. If there are multiple correct answers, print any of them.

Examples

input

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

output

```
3
1 2 3
```

input

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

output

```
-1
```

input

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

output

```
1
1
```

Note

In the first example $k = 2$ and Olya has three cartons with expiry dates 0, 1 and 1 (they expire today, tomorrow and tomorrow), and the shop has 3 cartons with expiry date 0 and 3 cartons with expiry date 2. Olya can buy three cartons, for example, one with the expiry date 0 and two with expiry date 2.

In the second example all three cartons Olya owns expire today and it means she would have to throw packets away regardless of whether she buys an extra one or not.

In the third example Olya would drink $k = 2$ cartons today (one she alreay has in her fridge and one from the shop) and the remaining one tomorrow.

E. Change-free

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

Student Arseny likes to plan his life for n days ahead. He visits a canteen every day and he has already decided what he will order in each of the following n days. Prices in the canteen do not change and that means Arseny will spend c_i rubles during the i-th day.

There are 1-ruble coins and 100-ruble notes in circulation. At this moment, Arseny has m coins and a sufficiently large amount of notes (you can assume that he has an infinite amount of them). Arseny loves modern technologies, so he uses his credit card everywhere except the canteen, but he has to pay in cash in the canteen because it does not accept cards.

Cashier always asks the student to pay change-free. However, it's not always possible, but Arseny tries to minimize the *dissatisfaction* of the cashier. Cashier's dissatisfaction for each of the days is determined by the total amount of notes and coins in the change. To be precise, if the cashier gives Arseny x notes and coins on the i-th day, his dissatisfaction for this day equals $x \cdot w_i$. Cashier always gives change using as little coins and notes as possible, he always has enough of them to be able to do this.

"Caution! Angry cashier"

Arseny wants to pay in such a way that the total dissatisfaction of the cashier for n days would be as small as possible. Help him to find out how he needs to pay in each of the n days!

Note that Arseny always has enough money to pay, because he has an infinite amount of notes. Arseny can use notes and coins he received in change during any of the following days.

Input

The first line contains two integers n and m ($1 \le n \le 10^5$, $0 \le m \le 10^9$) — the amount of days Arseny planned his actions for and the amount of coins he currently has.

The second line contains a sequence of integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 10^5$) — the amounts of money in rubles which Arseny is going to spend for each of the following days.

The third line contains a sequence of integers $w_1, w_2, \ldots, w_n$ ($1 \le w_i \le 10^5$) — the cashier's dissatisfaction coefficients for each of the following days.

Output

In the first line print one integer — minimum possible total dissatisfaction of the cashier.

Then print n lines, the i-th of then should contain two numbers — the amount of notes and the amount of coins which Arseny should use to pay in the canteen on the i-th day.

Of course, the total amount of money Arseny gives to the casher in any of the days should be no less than the amount of money he has planned to spend. It also shouldn't exceed 10^6 rubles: Arseny never carries large sums of money with him.

If there are multiple answers, print any of them.

Examples

input

```
5 42
117 71 150 243 200
1 1 1 1 1
```

output

```
79
1 17
1 0
2 0
2 43
2 0
```

input

```
3 0
100 50 50
1 3 2
```

output

```
150
1 0
1 0
0 50
```

input

```
5 42
117 71 150 243 200
5 4 3 2 1
```

output

```
230
1 17
1 0
1 50
3 0
2 0
```