D. Event Dates

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

On a history lesson the teacher asked Vasya to name the dates when n famous events took place. He doesn't remembers the exact dates but he remembers a segment of days $[l_i, r_i]$ (inclusive) on which the event could have taken place. However Vasya also remembers that there was at most one event in one day. Help him choose such n dates of famous events that will fulfill both conditions. It is guaranteed that it is possible.

Input

The first line contains one integer n $(1 \le n \le 100)$ — the number of known events. Then follow n lines containing two integers l_i and r_i each $(1 \le l_i \le r_i \le 10^7)$ — the earliest acceptable date and the latest acceptable date of the i-th event.

Output

Print n numbers — the dates on which the events took place. If there are several solutions, print any of them. It is guaranteed that a solution exists.

Sample test(s)

input

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

output

```
1 2 3
```

input

```
2
1 3
1 3
```

output

```
1 2
```

B. Discounts

time limit per test: 3 seconds

memory limit per test: 256 megabytes

One day Polycarpus stopped by a supermarket on his way home. It turns out that the supermarket is having a special offer for stools. The offer is as follows: if a customer's shopping cart contains at least one stool, the customer gets a 50% discount on the cheapest item in the cart (that is, it becomes two times cheaper). If there are several items with the same minimum price, the discount is available for only one of them!

Polycarpus has k carts, and he wants to buy up all stools and pencils from the supermarket. Help him distribute the stools and the pencils among the shopping carts, so that the items' total price (including the discounts) is the least possible.

Polycarpus must use all k carts to purchase the items, no shopping cart can remain empty. Each shopping cart can contain an arbitrary number of stools and/or pencils.

Input

The first input line contains two integers n and k ($1 \le k \le n \le 10^3$) — the number of items in the supermarket and the number of carts, correspondingly. Next n lines describe the items as "c_i t_i" (without the quotes), where c_i ($1 \le c_i \le 10^9$) is an integer denoting the price of the i-th item, t_i ($1 \le t_i \le 2$) is an integer representing the type of item i (1 for a stool and 2 for a pencil). The numbers in the lines are separated by single spaces.

Output

In the first line print a single real number **with exactly one** decimal place — the minimum total price of the items, including the discounts.

In the following k lines print the descriptions of the items in the carts. In the i-th line print the description of the i-th cart as "t b_1 b_2 ... b_t" (without the quotes), where t is the number of items in the i-th cart, and the sequence $b_1, b_2, ..., b_t$ ($1 \le b_j \le n$) gives the indices of items to put in this cart in the optimal distribution. All indices of items in all carts should be pairwise different, each item must belong to exactly one cart. You can print the items in carts and the carts themselves in any order. The items are numbered from 1 to n in the order in which they are specified in the input.

If there are multiple optimal distributions, you are allowed to print any of them.

Sample test(s)

input

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

output

```
5.5
2 1 2
1 3
```

input

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

output

```
8.0
1 1
2 4 2
1 3
```

Note

In the first sample case the first cart should contain the 1st and 2nd items, and the second cart should contain the 3rd item. This way each cart has a stool and each cart has a 50% discount for the cheapest item. The total price of all items will be: $2 \cdot 0.5 + (3 + 3 \cdot 0.5) = 1 + 4.5 = 5.5$.

C. Deletion of Repeats

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

Once Bob saw a string. It contained so many different letters, that the letters were marked by numbers, but at the same time each letter could be met in the string at most 10 times. Bob didn't like that string, because it contained repeats: a repeat of length x is such a substring of length $2x$, that its first half coincides character by character with its second half. Bob started deleting all the repeats from the string. He does it as follows: while it's possible, Bob takes the shortest repeat, if it is not unique, he takes the leftmost one, and deletes its left half and everything that is to the left of this repeat.

You're given the string seen by Bob. Find out, what it will look like after Bob deletes all the repeats in the way described above.

Input

The first input line contains integer n $(1 \le n \le 10^5)$ — length of the string. The following line contains n space-separated integer numbers from 0 to 10^9 inclusive — numbers that stand for the letters of the string. It's guaranteed that each letter can be met in the string at most 10 times.

Output

In the first line output the length of the string's part, left after Bob's deletions. In the second line output all the letters (separated by a space) of the string, left after Bob deleted all the repeats in the described way.

Sample test(s)

input

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

output

```
3
1 2 3
```

input

```
7
4 5 6 5 6 7 7
```

output

```
1
7
```

A. Petya and Inequiations

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

Little Petya loves inequations. Help him find n positive integers $a_1, a_2, ..., a_n$, such that the following two conditions are satisfied:

- $a_1{}^2 + a_2{}^2 + ... + a_n{}^2 \ge x$
- $a_1 + a_2 + ... + a_n \le y$

Input

The first line contains three space-separated integers n, x and y ($1 \le n \le 10^5$, $1 \le x \le 10^{12}$, $1 \le y \le 10^6$).

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

Output

Print n positive integers that satisfy the conditions, one integer per line. If such numbers do not exist, print a single number "-1". If there are several solutions, print any of them.

Sample test(s)

input

```
5 15 15
```

output

```
4
4
1
1
2
```

input

```
2 3 2
```

output

```
-1
```

input

```
1 99 11
```

output

```
11
```

B. Lucky Common Subsequence

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

In mathematics, a *subsequence* is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements. For example, the sequence BDF is a subsequence of ABCDEF. A *substring* of a string is a continuous subsequence of the string. For example, BCD is a substring of ABCDEF.

You are given two strings s_1, s_2 and another string called $virus$. Your task is to find the longest common subsequence of s_1 and s_2, such that it doesn't contain $virus$ as a substring.

Input

The input contains three strings in three separate lines: s_1, s_2 and $virus$ ($1 \le |s_1|, |s_2|, |virus| \le 100$). Each string consists only of uppercase English letters.

Output

Output the longest common subsequence of s_1 and s_2 without $virus$ as a substring. If there are multiple answers, any of them will be accepted.

If there is no valid common subsequence, output 0.

Sample test(s)

input

```
AJKEQSLOBSROFGZ
OVGURWZLWVLUXTH
OZ
```

output

```
ORZ
```

input

```
AA
A
A
```

output

```
0
```

12367. Designing T-Shirts

Problem code: TAP2012D

[The original version of this problem (in Spanish) can be found at http://www.dc.uba.ar/events/icpc/download/problems/taip2012-problems.pdf]

Argentina's rugby is currently in one of its best moments of all time. Recently the under-18 and under-21 national teams qualified for their corresponding world cups, so the coaches of both teams have asked the Incredible Commission for the Production of Clothing (ICPC) to provide the t-shirts for these events. Each team is formed by **N** players, but because the two world cups do not take place simultaneously it was agreed that the ICPC would provide only **N** t-shirts, to be used by both teams.

For this reason, the t-shirts must be a valid set of clothing for both teams. The rules of the rugby world cups state that each player must go in the field with a t-shirt imprinted with a unique number, along with a prefix of the player's surname, not necessarily unique. This includes boundary cases such as a t-shirt with no surname prefix (that is, a prefix of length **0**) and a t-shirt with a complete surname.

The experts of ICPC immediately realized that they could simply provide **N** t-shirts with only numbers and no surnames on them, and each of them would be a valid t-shirt to be used by any player of any of the two teams. However, the coaches would rather have the t-shirts with the longest possible prefixes, of course without violating world cup rules, because this way it's easier for them to identify the players while the matches are taking place.

Your task is to help the ICPC finding the maximum amount of letters that can be imprinted on a set of **N** t-shirts, so that this set is a valid clothing set for both teams. For example, if we have **N = 3** players, the under-18 team is composed of `"PEREZ"`, `"GONZALEZ"` and `"LOPEZ"`, whereas the under-21 team is composed of `"GARCIA"`, `"PERALTA"` and `"RODRIGUEZ"`, the optimal choice consists in having one t-shirt with the **1**-letter prefix `"G"` (to be used by `"GONZALEZ"` and `"GARCIA"`), anotherone with the **3**-letter prefix `"PER"` (to be used by `"PEREZ"` and `"PERALTA"`), and the third t-shirt with a **0**-letter prefix (to be used by `"LOPEZ"` and `"RODRIGUEZ"`). This way, the answer in this case would be **1+3+0=4**.

Input

Each test case is described using three lines. The first line contains a single integer number **N**, indicating the number of players in each of the two teams (**$1 \leq N \leq 10^4$**). The second line contains the surnames of the **N** players in the under-18 team, whereas the third line contains the surnames of the **N** players in the under-21 team. Each surname is a non-empty string of at most **100** uppercase letters. In each test case the total number of characters in the **2N** surnames is at most **10^5**, and two or more players of the same or different teams may have the same surname.

The end of the input is indicated by a line containing the number **-1**.

Output

For each test case, you should print a single line containing an integer number, representing the maximum number of letters that can be imprinted on a set of **N** valid t-shirts to be used by both teams as explained in the problem statement.

Example

```
Input:
3
PEREZ GONZALEZ LOPEZ
GARCIA PERALTA RODRIGUEZ
2
RODRIGO GONZALEZ
GONZALO RODRIGUEZ
3
LOPEZ PEREZ LOPEZ
PEREZ LOPEZ LOPEZ
1
GIMENEZ
JIMENEZ
6
HEIDEGGER GAUSS GROTHENDIECK ERDOS CHURCH TURING
HEISENBERG GALOIS EULER ALLEN GODEL CHURCHILL
-1

Output:
4
12
15
0
13
```

A. Guilty — to the kitchen!

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

It's a very unfortunate day for Volodya today. He got bad mark in algebra and was therefore forced to do some work in the kitchen, namely to cook borscht (traditional Russian soup). This should also improve his algebra skills.

According to the borscht recipe it consists of n ingredients that have to be mixed in proportion $(a_1 : a_2 : \ldots : a_n)$ litres (thus, there should be $a_1 \cdot x$, ..., $a_n \cdot x$ litres of corresponding ingredients mixed for some non-negative x). In the kitchen Volodya found out that he has b_1, ..., b_n litres of these ingredients at his disposal correspondingly. In order to correct his algebra mistakes he ought to cook as much soup as possible in a V litres volume pan (which means the amount of soup cooked can be between 0 and V litres). What is the volume of borscht Volodya will cook ultimately?

Input

The first line of the input contains two space-separated integers n and V ($1 \le n \le 20$, $1 \le V \le 10000$). The next line contains n space-separated integers a_i ($1 \le a_i \le 100$). Finally, the last line contains n space-separated integers b_i ($0 \le b_i \le 100$).

Output

Your program should output just one real number — the volume of soup that Volodya will cook. Your answer must have a relative or absolute error less than 10^{-4}.

Sample test(s)

input

```
1 100
1
40
```

output

```
40.0
```

input

```
2 100
1 1
25 30
```

output

```
50.0
```

input

```
2 100
1 1
60 60
```

output

```
100.0
```

D. Mr. Bender and Square

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

Mr. Bender has a digital table of size $n \times n$, each cell can be switched on or off. He wants the field to have at least c switched on squares. When this condition is fulfilled, Mr Bender will be happy.

We'll consider the table rows numbered from top to bottom from 1 to n, and the columns — numbered from left to right from 1 to n. Initially there is exactly one switched on cell with coordinates (x, y) (x is the row number, y is the column number), and all other cells are switched off. Then each second we switch on the cells that are off but have the side-adjacent cells that are on.

For a cell with coordinates (x, y) the side-adjacent cells are cells with coordinates $(x - 1, y)$, $(x + 1, y)$, $(x, y - 1)$, $(x, y + 1)$.

In how many seconds will Mr. Bender get happy?

Input

The first line contains four space-separated integers n, x, y, c ($1 \le n, c \le 10^9$; $1 \le x, y \le n$; $c \le n^2$).

Output

In a single line print a single integer — the answer to the problem.

Sample test(s)

input

```
6 4 3 1
```

output

```
0
```

input

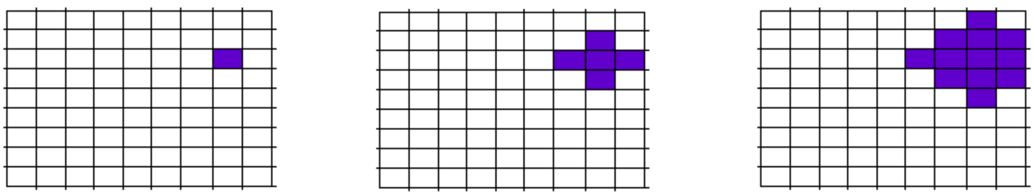
```
9 3 8 10
```

output

```
2
```

Note

Initially the first test has one painted cell, so the answer is 0. In the second test all events will go as is shown on the figure.



.

A. Cut Ribbon

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

Polycarpus has a ribbon, its length is n. He wants to cut the ribbon in a way that fulfils the following two conditions:

- After the cutting each ribbon piece should have length a, b or c.
- After the cutting the number of ribbon pieces should be maximum.

Help Polycarpus and find the number of ribbon pieces after the required cutting.

Input

The first line contains four space-separated integers n, a, b and c $(1 \le n, a, b, c \le 4000)$ — the length of the original ribbon and the acceptable lengths of the ribbon pieces after the cutting, correspondingly. The numbers a, b and c can coincide.

Output

Print a single number — the maximum possible number of ribbon pieces. It is guaranteed that at least one correct ribbon cutting exists.

Sample test(s)

input

```
5 5 3 2
```

output

```
2
```

input

```
7 5 5 2
```

output

```
2
```

Note

In the first example Polycarpus can cut the ribbon in such way: the first piece has length 2, the second piece has length 3.

In the second example Polycarpus can cut the ribbon in such way: the first piece has length 5, the second piece has length 2.

12372. In Debt

Problem code: TAP2012I

[The original version of this problem (in Spanish) can be found at http://www.dc.uba.ar/events/icpc/download/problems/taip2012-problems.pdf]

Ignacio and Ines really like science. Luckily they live in Nlogonia, where as everyone knows there are **N** science museums. Both Ignacio and Ines have the next **N** Saturdays free, so they have agreed on a schedule to visit a different science museum each of these days.

Ignacio is quite stingy, so every Saturday he will tell Ines that he has forgotten to bring the money to pay the museum's entrance fee, and will therefore ask her to pay for him. Ines always does so, and because she knows him well she also knows that unless she asks for her money back, he will never pay up. In fact, Ines is certain that even if she asks Ignacio for her money back, he will only accept to pay if the cumulative debt is a multiple of **100**, because otherwise he will argue that he has no change to pay exactly, and will thus pay nothing at all.

This being the situation, every Sunday if the cumulative debt is a multiple of **100** Ines will go to Ignacio's house to claim her money, and because he'll have no excuse left, he will pay without any type of protest. Of course he doesn't like this, but he is comforted by the idea that if the cumulative debt after visiting the **N** museums is not a multiple of **100**, Ines shall not claim the last part of her money.

Ines would like to know how many times she will have to go to Ignacio's house to ask for her money. In order to calculate this, she can provide a list of the prices of the entrance fees to the **N** science museums in Nlogonia, in the order that she and Ignacio are going to visit them.

Input

Each test case is described using two lines. The first line contains a single integer number **N**, indicating the number of science museums in Nlogonia (**1 ≤ N ≤ 100**). The second line contains **N** integers P_i representing the prices of the entrance fees to the different museums, in the order in which they are going to be visited (**1 ≤** P_i **≤ 100** for **i = 1, ..., N**). The end of the input is signalled by a line containing the number **-1**.

Output

For each test case, you should print a single line containing an integer number, representing the number of times Ines is going to have to go by Ignacio's house to ask for her money.

Example

```
Input:
3
50 50 50
5
50 100 100 100 100
9
25 50 75 100 25 50 75 100 25
5
35 45 20 22 33
3
100 100 100
-1

Output:
1
0
2
1
3
```

show comments

D. Road Map

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

There are n cities in Berland. Each city has its index — an integer number from 1 to n. The capital has index r_1. All the roads in Berland are two-way. The road system is such that there is exactly one path from the capital to each city, i.e. the road map looks like a tree. In Berland's chronicles the road map is kept in the following way: for each city i, different from the capital, there is kept number p_i — index of the last city on the way from the capital to i.

Once the king of Berland Berl XXXIV decided to move the capital from city r_1 to city r_2. Naturally, after this the old representation of the road map in Berland's chronicles became incorrect. Please, help the king find out a new representation of the road map in the way described above.

Input

The first line contains three space-separated integers n, r_1, r_2 ($2 \le n \le 5 \cdot 10^4$, $1 \le r_1 \ne r_2 \le n$) — amount of cities in Berland, index of the old capital and index of the new one, correspondingly.

The following line contains $n - 1$ space-separated integers — the old representation of the road map. For each city, apart from r_1, there is given integer p_i — index of the last city on the way from the capital to city i. All the cities are described in order of increasing indexes.

Output

Output $n - 1$ numbers — new representation of the road map in the same format.

Sample test(s)

input

```
3 2 3
2 2
```

output

```
2 3
```

input

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

output

```
6 4 1 4 2
```