Codeforces Beta Round #34 (Div. 2)

A. Reconnaissance 2

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

n soldiers stand in a circle. For each soldier his height a_i is known. A reconnaissance unit can be made of such two **neighbouring** soldiers, whose heights difference is minimal, i.e. $|a_i - a_j|$ is minimal. So each of them will be less noticeable with the other. Output any pair of soldiers that can form a reconnaissance unit.

Input

The first line contains integer n ($2 \le n \le 100$) — amount of soldiers. Then follow the heights of the soldiers in their order in the circle — n space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 1000$). The soldier heights are given in clockwise or counterclockwise direction.

Output

Output two integers — indexes of **neighbouring** soldiers, who should form a reconnaissance unit. If there are many optimum solutions, output any of them. Remember, that the soldiers stand in a circle.

Examples

input
5 10 12 13 15 10
output
5 1

input
4 10 20 30 40
output
1 2

B. Sale

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

Once Bob got to a sale of old TV sets. There were n TV sets at that sale. TV set with index i costs a_i bellars. Some TV sets have a negative price — their owners are ready to pay Bob if he buys their useless apparatus. Bob can «buy» any TV sets he wants. Though he's very strong, Bob can carry at most m TV sets, and he has no desire to go to the sale for the second time. Please, help Bob find out the maximum sum of money that he can earn.

Input

The first line contains two space-separated integers n and m ($1 \le m \le n \le 100$) — amount of TV sets at the sale, and amount of TV sets that Bob can carry. The following line contains n space-separated integers a_i ($-1000 \le a_i \le 1000$) — prices of the TV sets.

Output

Output the only number — the maximum sum of money that Bob can earn, given that he can carry at most m TV sets.

Examples

input
5 3 -6 0 35 -2 4
output
8

input
4 2 7 0 0 -7
output
7

C. Page Numbers

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

«Bersoft» company is working on a new version of its most popular text editor — Bord 2010. Bord, like many other text editors, should be able to print out multipage documents. A user keys a sequence of the document page numbers that he wants to print out (separates them with a comma, without spaces).

Your task is to write a part of the program, responsible for «standardization» of this sequence. Your program gets the sequence, keyed by the user, as input. The program should output this sequence in format l_1-r_1,l_2-r_2,...,l_k-r_k, where $r_i + 1 < l_{i+1}$ for all i from 1 to $k - 1$, and $l_i \le r_i$. The new sequence should contain all the page numbers, keyed by the user, and nothing else. If some page number appears in the input sequence several times, its appearances, starting from the second one, should be ignored. If for some element i from the new sequence $l_i = r_i$, this element should be output as l_i, and not as «l_i - l_i».

For example, sequence `1,2,3,1,1,2,6,6,2` should be output as `1-3,6`.

Input

The only line contains the sequence, keyed by the user. The sequence contains at least one and at most 100 positive integer numbers. It's guaranteed, that this sequence consists of positive integer numbers, not exceeding 1000, separated with a comma, doesn't contain any other characters, apart from digits and commas, can't end with a comma, and the numbers don't contain leading zeroes. Also it doesn't start with a comma or contain more than one comma in a row.

Output

Output the sequence in the required format.

Examples

input
1,2,3,1,1,2,6,6,2
output
1-3,6

input
3,2,1
output
1-3

input
30,20,10
output
10,20,30

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.

Examples

input
3 2 3 2 2
output
2 3

input
6 2 4 6 1 2 4 2
output
6 4 1 4 2

E. Collisions

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

On a number line there are n balls. At time moment 0 for each ball the following data is known: its coordinate x_i, speed v_i (possibly, negative) and weight m_i. The radius of the balls can be ignored.

The balls collide elastically, i.e. if two balls weighing m_1 and m_2 and with speeds v_1 and v_2 collide, their new speeds will be:

$$v_1' = \frac{(m_1 - m_2)v_1 + 2m_2v_2}{m_1 + m_2}, v_2' = \frac{(m_2 - m_1)v_2 + 2m_1v_1}{m_1 + m_2}$$

.
Your task is to find out, where each ball will be t seconds after.

Input

The first line contains two integers n and t ($1 \le n \le 10, 0 \le t \le 100$) — amount of balls and duration of the process. Then follow n lines, each containing three integers: x_i, v_i, m_i ($1 \le |v_i|, m_i \le 100, |x_i| \le 100$) — coordinate, speed and weight of the ball with index i at time moment 0.

It is guaranteed that no two balls have the same coordinate initially. Also each collision will be a collision of not more than two balls (that is, three or more balls never collide at the same point in all times from segment $[0;t]$).

Output

Output n numbers — coordinates of the balls t seconds after. Output the numbers accurate to at least 4 digits after the decimal point.

Examples

input
2 9 3 4 5 0 7 8
output
68.538461538 44.538461538

input
3 10 1 2 3 4 -5 6 7 -8 9
output
-93.666666667 -74.666666667 -15.666666667