



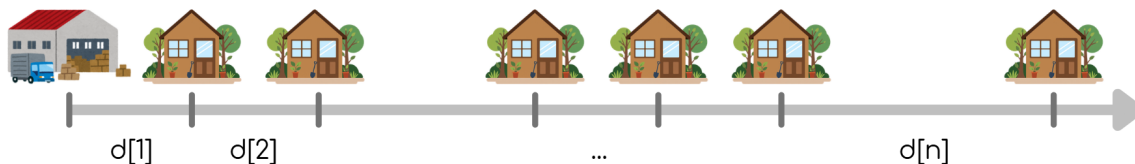
Task 2: Famished Cats

After successfully preventing the extinction of cats during the annual National Cat Day (NCD) celebration, Ket the cat has now received hunger complaints from the kingdom of cannibalistic cats. Hence, Ket has been tasked with delivering food to prevent them from resorting to cannibalism again.

This cat kingdom can be modelled as a very long road that runs from west to east. There is a food bank at the west end of the road. There are n cat houses to the east of the food bank, numbered from 1 to n . It is guaranteed that n is **even**. The first house is $d[1]$ kilometres east of the food bank. For $i \geq 2$, the i -th house is $d[i]$ kilometres east of the $(i - 1)$ -th house.

Ket will drive a delivery truck to deliver food to the houses. The truck starts from the food bank and initially has x units of fuel. 1 unit of fuel allows Ket to drive his truck 1 kilometre along the road.

The i -th house has $f[i]$ units of fuel for the truck to use. The truck has infinite fuel storage and stops only when it runs out of fuel. The truck does not need to return to the food bank after leaving.



In addition, Ket has a magic wand. In one wave of his wand, he can **swap** the amount of fuel stored in the i -th and $(n - i + 1)$ -th cats' houses. This swap can only take place if the fuels in both houses have not been used yet. Help Ket find the index of the farthest house D he can reach, using any number of swaps. Also, help Ket find the minimum number of swaps S required to reach house D .

You may refer to Sample Test Case 1 for a more detailed explanation.

Input format

Your program must read from standard input.

The first line of input contains two space-separated integers n and x .

The second line of input contains n space-separated integers $d[1], d[2], \dots, d[n]$.

The third line of input contains n space-separated integers $f[1], f[2], \dots, f[n]$.



Output format

Your program must print to standard output.

Output **two** space-separated integers on a single line. The first integer should be D , the index of the furthest house Ket can reach, followed by S , the minimum number of swaps required to reach house D .

Subtasks

For all test cases, the input will satisfy the following bounds:

- $2 \leq n \leq 500\,000$
- n is even.
- $1 \leq d[i] \leq 10^9$ for all $1 \leq i \leq n$
- $1 \leq f[i] \leq 10^9$ for all $1 \leq i \leq n$
- $d[1] \leq x \leq 10^9$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Score	Additional constraints
0	0	Sample test cases
1	7	$ f[i] - f[n - i + 1] = k$ for some constant k , for all $1 \leq i \leq n$
2	12	$n \leq 40$
3	14	f is non-decreasing ($f[i] \leq f[i + 1]$ for all $1 \leq i \leq n - 1$)
4	19	$D \leq \frac{n}{2}$
5	21	$n \leq 5000$
6	27	No additional constraints

Note: The absolute value of a number x , denoted $|x|$, is the non-negative number equal to the distance between 0 and x on a number line. For example, $|5| = 5$ and $|-5| = 5$.

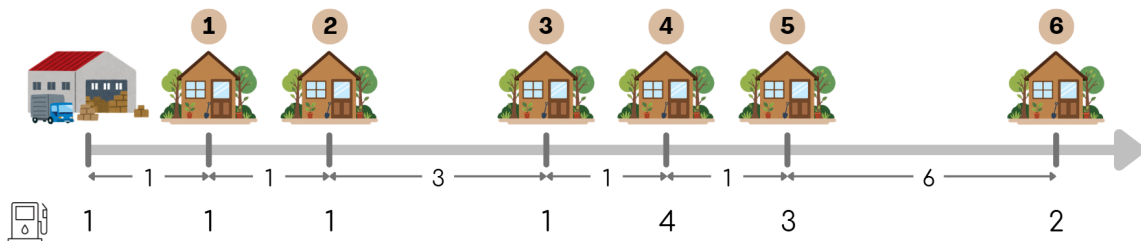


Sample Test Case 1

This test case is valid for subtasks 2, 5, and 6.

Input	Output
<pre>6 1 1 1 3 1 1 6 1 1 1 4 3 2</pre>	<pre>5 1</pre>

Sample Test Case 1 Explanation



There is a food bank with $n = 6$ houses to its east. Ket's truck starts with $x = 1$ unit of fuel and moves to house 1. He then refuels at house 1 and drives to house 2. At this point, it is optimal for Ket to use his magic wand to swap the fuel amounts in houses 2 and 5, allowing him to refill 3 units of fuel and reach house 3. Subsequently, he can refill 1 unit of fuel before travelling to the next two houses, refilling 4 and 1 units (due to the earlier swap) of fuel in houses 4 and 5 respectively.

He will then have 4 units of fuel left, leaving him unable to travel to house 6 as it requires 6 units of fuel. Since Ket runs out of fuel before reaching house 6, he can only travel to house 5. Furthermore, he had to make one swap with his magic wand. Therefore, $D = 5$ and $S = 1$.

Sample Test Case 2

This test case is valid for subtasks 1, 2, 5, and 6.

Input	Output
<pre>6 5 3 8 3 1 4 1 2 7 1 6 2 7</pre>	<pre>6 1</pre>



Sample Test Case 3

This test case is valid for subtasks 2, 3, 4, 5, and 6.

Input	Output
6 2 2 24 25 40 5 11 4 12 14 16 20 30	3 2

Sample Test Case 4

This test case is valid for subtasks 2, 5, and 6.

Input	Output
6 10 3 6 3 7 8 6 4 3 1 7 1 6	5 1