

H. Stock Exchange

time limit per test: 6 seconds
 memory limit per test: 16 megabytes
 input: standard input
 output: standard output

Warning: This problem has an unusual memory limit!

Bob decided that he will not waste his prime years implementing GUI forms for a large corporation and instead will earn his supper on the Stock Exchange Reykjavik. The Stock Exchange Reykjavik is the only actual stock exchange in the world. The only type of transaction is to take a single share of stock x and exchange it for a single share of stock y , provided that the current price of share x is at least the current price of share y .

There are $2n$ stocks listed on the SER that are of interest to Bob, numbered from 1 to $2n$. Bob owns a single share of stocks 1 through n and would like to own a single share of each of $n + 1$ through $2n$ some time in the future.

Bob managed to forecast the price of each stock — in time $t \geq 0$, the stock i will cost $a_i \cdot \lfloor t \rfloor + b_i$. The time is currently $t = 0$. Help Bob find the earliest moment in time in which he can own a single share of each of $n + 1$ through $2n$, and the minimum number of stock exchanges he has to perform in order to do that.

You may assume that the Stock Exchange has an unlimited amount of each stock at any point in time.

Input

The first line contains a single integer n ($1 \leq n \leq 2200$) — the number stocks currently owned by Bob.

Each of the next $2n$ lines contains integers a_i and b_i ($0 \leq a_i, b_i \leq 10^9$), representing the stock price of stock i .

Output

If it is impossible for Bob to achieve his goal, output a single integer -1 .

Otherwise, output two integers T and E , where T is the minimum time in which he can achieve his goal, and E is the minimum number of exchanges in which he can achieve his goal at time T .

Examples

input	Copy
1 3 10 1 16	
output	Copy
3 1	

input	Copy
2 3 0 2 1 1 10 1 11	
output	Copy

Codeforces Global Round 4

Contest is running

00:44:20

Contestant



→ Submit?

Language: Python 3.7.2
 Almost always, if you send a solution on PyPy, it works much faster

Choose file: [Choose File](#) No file chosen

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

[Submit](#)

→ Score table

	Score
Problem A	332
Problem B	498
Problem C	830
Problem D	1162
Problem E	1328
Problem F1	996
Problem F2	996
Problem G	2158
Problem H	2656
Successful hack	100
Unsuccessful hack	-50
Unsuccessful submission	-50
Resubmission	-50

* If you solve problem on 01:45 from the first attempt

6 3

input

Copy

1
42 42
47 47

output

Copy

-1

input

Copy

8
145 1729363
891 4194243
424 853731
869 1883440
843 556108
760 1538373
270 762781
986 2589382
610 99315884
591 95147193
687 99248788
65 95114537
481 99071279
293 98888998
83 99764577
769 96951171

output

Copy

434847 11

input

Copy

8
261 261639
92 123277
271 131614
320 154417
97 258799
246 17926
117 222490
110 39356
85 62864876
86 62781907
165 62761166
252 62828168
258 62794649
125 62817698
182 62651225
16 62856001

output

Copy

1010327 11

Note

In the first example, Bob simply waits until time $t = 3$, when both stocks cost exactly the same amount.

In the second example, the optimum strategy is to exchange stock 2 for stock 1 at time $t = 1$, then exchange one share of stock 1 for stock 3 at time $t = 5$ (where both cost 15) and then at time $t = 6$ exchange the second one for the stock number 4 (when they cost 18 and 17, respectively). Note that he can achieve his goal also with only two exchanges, but this would take total time of $t = 9$, when he would finally be able to exchange the share number 2 for the share number 3.

In the third example, Bob can never achieve his goal, as the second stock is always strictly more expensive than the first one.

The only programming contests Web 2.0 platform

Server time: Jul/20/2019 22:50:35^{UTC+5.5} (e1).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#).

Supported by



ITMO UNIVERSITY