Problem F. Election

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	256 mebibytes

An election was held today. A total of n parties, numbered 1 through n, has participated in this election, and m slots were distributed among the parties based on the number of votes each party got. The following algorithm was used for slot distribution:

Suppose that the parties $1, 2, \ldots, n$ got $c_1, c_2, \ldots, c_n$ votes, respectively. Let $s = c_1 + c_2 + \ldots + c_n$. First, for each i, $\lfloor \frac{c_i}{s} \cdot m \rfloor$ slots are distributed to the party i. Then, the remaining slots are distributed from the parties with the larger value of the fractional part of $\frac{c_i}{s} \cdot m$, one slot per party. In case of a tie, the lower-indexed party has the priority.

You have the following information:

- The parties $1, 2, \ldots, n$ got exactly $a_1, a_2, \ldots, a_n$ votes, respectively.
- The parties $1, 2, \ldots, n$ got at least $b_1, b_2, \ldots, b_n$ slots, respectively.

Compute the minimum possible number of total slots m.

Input

The first line of input contains one integer n ($1 \le n \le 100$). Then n lines follow, each contains a pair of integers a_i and b_i ($1 \le a_i \le 1000$, $0 \le b_i \le 10^9$). You may assume that there exists at least one i such that $b_i \ge 1$.

Output

Print the minimum possible number of total slots m.

Examples

standard input	standard output
3 1 2 4 5 2 3	11
4 1 0 6 5 4 4 5 8	25
1 42 42	42