TASK	KORNISLAV	RESETO	PERKET	SVADA	SETNJA	CAVLI
input	standard input					
output	standard output					
time limit	1 second	1 second	1 second	1 second	1 second	2 seconds
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
points	**30**	**40**	**70**	**100**	**120**	**140**
	500					

Kornislav the turtle never has anything interesting to do. Since he's going to live for three hundred years, he keeps trying to find way to kill time. This weekend he started playing "enclose the largest rectangle".

To start with, Kornislav needs four positive integers. He tries to enclose a rectangle by moving in one direction, then turning 90 degrees, then walking in the new direction etc. Kornislav makes a total of three 90-degree turns and walks four segments.

When walking in some direction, the number of steps he takes must be equal to one of the four chosen integers and each integer must be used exactly once. Depending on the order in which Kornislav uses the integers, his walk will make various shapes, some of which don't contain enclosed rectangles.

Write a program that calculates the largest rectangle the turtle can enclose with its walk.

INPUT

The first line contains four positive integers A, B, C and D ($0 < A, B, C, D < 100$), the four chosen integers.

OUTPUT

Output the largest area.

EXAMPLES

input 1 2 3 4 **output** 3	**input** 4 4 3 4 **output** 12

In the first example, one possible way for Kornislav to enclose a rectangle of area 3:

- Make 4 steps forward;
- Turn right;
- Make 1 step forward;
- Turn right;
- Make 3 steps forward;
- Turn right;
- Make 2 steps forward.

The sieve of Eratosthenes is a famous algorithm to find all prime numbers up to N. The algorithm is:

1. Write down all integers between 2 and N, inclusive.
2. Find the **smallest** number not already crossed out and call it P; P is prime.
3. Cross out P and all its multiples that **aren't already crossed out**.
4. If not all numbers have been crossed out, go to step 2.

Write a program that, given N and K, find the K-th integer to be crossed out.

INPUT

The integers N and K ($2 \le K < N \le 1000$).

OUTPUT

Output the K-th number to be crossed out.

EXAMPLES

input	**input**	**input**
`7 3`	`15 12`	`10 7`
output	**output**	**output**
`6`	`7`	`9`

In the third example, we cross out, in order, the numbers 2, 4, 6, 8, 10, 3, 9, 5 and 7. The seventh number is 9.

"Perket" is a widely known and delicious meal. For perket to be what it is, cooks must carefully choose the ingredients to get the fullest taste possible while keeping the meal traditional.

You have N ingredients at your disposal. For each we know its sourness S and bitterness B. When using multiple ingredients, the total sourness is the **product** of sourness amounts of all ingredients, while the total bitterness is the **sum** of bitterness amounts of all ingredients.

As everyone knows, perket is supposed to be neither sour nor bitter; we want to choose the ingredients so that the absolute difference between sourness and bitterness is the smallest.

Also, it is necessary to use at least one ingredient; you can't serve water as the main course.

INPUT

The first line contains the integer N ($1 \le N \le 10$), the number of ingredients at our disposal.

Each of the next N lines contains two integers separated by a space, the sourness and bitterness of each ingredient.

The input data will be such that, if we make a meal with all ingredients, both the sourness and bitterness will be less than 1 000 000 000.

OUTPUT

Output the smallest possible difference between sourness and bitterness.

EXAMPLES

input 1 3 10 **output** 7	**input** 2 3 8 5 8 **output** 1	**input** 4 1 7 2 6 3 8 4 9 **output** 1

In the third example, we choose the last three ingredients. The total sourness is then 2·3·4=24 and bitterness is 6+8+9=23. The difference is 1.

The local zoo has acquired a large open garden in which animals may freely move as in their natural habitats and entertain visitors with their usual shenanigans.

The most popular animals are monkeys. With their climbing and jumping and other skills, they delight old and young visitors alike.

One species of monkey has specialized in climbing tall trees and picking off coconuts. Another species has specialized in breaking them open.

There are N monkeys of the first type (numbered 1 through N) and M monkeys of the second type (numbered 1 through M).

Monkey k of the first type takes A_k seconds to find a good spot on the tree, after which it picks off its first coconut. After that the monkey produces a new coconut every B_k seconds.

Monkey k of the second type takes C_k seconds to find a good tool for opening the coconuts, after which it opens its first coconut. After that the monkey opens another coconut every D_k seconds.

Unfortunately, the second type of monkey is extremely aggressive so the two types may not be in the garden at the same time. Therefore, zoo keepers will chase away the first type of monkeys as soon as they have picked off all the coconuts. Similarly, if monkeys of the same type stay too long after opening all the coconuts, fights will ensue. Because of that, zoo keepers will send them away as soon as they have opened all the coconuts.

The zoo keepers first arrive immediately after all coconuts have been picked, and again immediately after the monkeys open them all. The time needed for monkeys to enter or leave the garden is also negligibly small.

Tomislav especially likes the second type of monkey, but can never guess when to arrive in order to see them. Help him calculate the time when the second type arrives if he knows the **total time that monkeys spent** in the garden, but **does not know the number of coconuts** in the garden.

INPUT

The first line contains the integer T ($1 \leq T \leq 1\,000\,000\,000$), the total time that monkeys spent in the garden, in seconds.

The next line contains the integer N ($1 \leq N \leq 100$), the number of monkeys of the first type.

Each of the following N lines contains two integers A_k and B_k ($1 \leq A_k, B_k \leq 1\,000\,000\,000$), how fast monkey k of the first type is.

The next line contains the integer M ($1 \leq M \leq 100$), the number of monkeys of the second type.

Each of the following M lines contains two integers C_k and D_k ($1 \leq C_k, D_k \leq 1\,000\,000\,000$), how fast monkey k of the second type is.

OUTPUT

Output the number of seconds between the arrival of the first type of monkeys and the arrival of the second type.

EXAMPLES

input 12 1 3 1 1 5 1 **output** 5	**input** 20 2 3 2 1 3 3 3 1 4 1 5 1 **output** 13

In the first example, it turns out there are three coconuts in the garden:

- The monkey of the first type picks off the first coconut 3 seconds after the garden was opened.
- The monkey picks off the second coconut 4 seconds after the garden is opened.
- The monkey picks off the third coconut 5 seconds after the garden is opened.
- Zoo keepers come in and escort the monkey out. The monkey of the second type arrives. The output is 5 because this is when Tomislav wants to arrive.
- The monkey of the second type opens the first coconut 10 seconds after the garden was opened.
- The monkey opens the second coconut 11 seconds after the garden was opened.
- The monkey opens the third coconut 12 seconds after the garden was opened.
- Zoo keepers come in and escort the monkey out.

In an infinite binary tree:

- Each node has exactly two children – a left and a right child.
- If a node is labeled with the integer X, then its left child is labeled 2·X and its right child 2·X+1.
- The root of the tree is labeled 1.

A **walk** on the binary tree starts in the root. Each step in the walk is either a jump onto the left child, onto the right child, or pause for rest (stay in the same node).

A walk is described with a string of letters 'L', 'R' and 'P':

- 'L' represents a jump to the left child;
- 'R' represents a jump to the right child;
- 'P' represents a pause.

The **value** of the walk is the label of the node we end up on. For example, the value of the walk LR is 5, while the value of the walk RPP is 3.

A **set** of walks is described by a string of characters 'L', 'R', 'P' and '*'. Each '*' can be any of the three moves; the set of walks contains all walks matching the pattern.

For example, the set L*R contains the walks LLR, LRR and LPR. The set ** contains the walks LL, LR, LP, RL, RR, RP, PL, PR and PP.

Finally, the **value of a set** of walks is the sum of values of all walks in the set.

Calculate the value of the given set of walks.

INPUT

A string describing the set. Only characters 'L', 'R', 'P' and '*' will appear and there will be at most 10000 of them.

OUTPUT

Output the value of the set.

SCORING

In test data worth 30% points, there will be no characters '*'.

In test data worth 50% points, there will be at most three characters '*'.

EXAMPLES

input	**input**	**input**	**input**
P*P	L*R	**	LLLLLRRRRRLLLLLRRRRRLLLLLRRRRRLLLLL
output	**output**	**output**	**output**
6	25	33	35400942560

Mirko found a wooden board and N nails in his attic. Mirko hammered the nails into the board as fast as possible. The board can be modeled by a coordinate plane and the nails as points in it. No two nails have the same x or the same y coordinate.

In order to keep having fun, Mirko stole his sister's elastic hair band, spread it over all nails and then let go. The elastic, naturally, tightened around the nails.

Mirko then repeats these steps while there are at least three nails in the board:

1. Write down the area of the shape enclosed by the hair band.
2. Picks the leftmost, rightmost, topmost or bottommost nail in the board.
3. Remove the chosen nail from the board; the elastic tightens again around the remaining nails.

Write a program that calculates the numbers written in step 1 of each iteration, if we know the nail Mirko picks in step 2 of each iteration.

INPUT

The first line contains the integer N (3 ≤ N ≤ 300000), the number of nails.

Each of the following N lines contains two integers separated by a space, the coordinates of a nail. All coordinates will be between 1 and 1000000000. No two nails will share the same x or y coordinate.

The next line contains N-2 letters 'L', 'R', 'U' or 'D'. The letters represent the nails Mirko picked in order:

- 'L' for the leftmost nail (smallest x coordinate),
- 'R' for the rightmost nail (largest x coordinate),
- 'U' for the topmost nail (largest y coordinate),
- 'D' for the bottommost nail (smallest y coordinate).

OUTPUT

Output N-2 numbers, each on a separate line. The numbers are, in order, the areas that Mirko wrote down. Output numbers with one digit after the decimal point.

SCORING

In test cases worth 50% points, N will be less than 1000.

EXAMPLES

input

```
5
1 4
2 2
4 1
3 5
5 3
LUR
```

output

```
9.0
6.5
2.5
```

input

```
8
1 6
2 4
3 1
4 2
5 7
6 5
7 9
8 3
URDLUU
```

output

```
34.0
24.0
16.5
14.0
9.5
5.0
```

The images below illustrate the state before each of the 6 steps in the second example.