A. Bicycle Chain

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

Vasya's bicycle chain drive consists of two parts: n stars are attached to the pedal axle, m stars are attached to the rear wheel axle. The chain helps to rotate the rear wheel by transmitting the pedal rotation.

We know that the i-th star on the pedal axle has a_i $(0 < a_1 < a_2 < \dots < a_n)$ teeth, and the j-th star on the rear wheel axle has b_j $(0 < b_1 < b_2 < \dots < b_m)$ teeth. Any pair (i, j) $(1 \le i \le n; 1 \le j \le m)$ is called a *gear* and sets the indexes of stars to which the chain is currently attached. Gear (i, j) has a gear ratio, equal to the value $\frac{b_j}{a_i}$.

Since Vasya likes integers, he wants to find such gears (i, j), that their ratios are integers. On the other hand, Vasya likes fast driving, so among all "integer" gears (i, j) he wants to choose a gear with the maximum ratio. Help him to find the number of such gears.

In the problem, fraction $\frac{b_j}{a_i}$ denotes division in real numbers, that is, no rounding is performed.

Input

The first input line contains integer n $(1 \le n \le 50)$ — the number of stars on the bicycle's pedal axle. The second line contains n integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 10^4)$ in the order of strict increasing.

The third input line contains integer m $(1 \le m \le 50)$ — the number of stars on the rear wheel axle. The fourth line contains m integers $b_1, b_2, \dots, b_m$ $(1 \le b_i \le 10^4)$ in the order of strict increasing.

It is guaranteed that there exists at least one gear (i, j), that its gear ratio is an integer. The numbers on the lines are separated by spaces.

Output

Print the number of "integer" gears with the maximum ratio among all "integer" gears.

Examples

input

```
2
4 5
3
12 13 15
```

output

```
2
```

input

```
4
1 2 3 4
5
10 11 12 13 14
```

output

```
1
```

Note

In the first sample the maximum "integer" gear ratio equals 3. There are two gears that have such gear ratio. For one of them $a_1 = 4, b_1 = 12$, and for the other $a_2 = 5, b_3 = 15$.

B. Olympic Medal

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

The World Programming Olympics Medal is a metal disk, consisting of two parts: the first part is a ring with outer radius of r_1 cm, inner radius of r_2 cm, $(0 < r2 < r1)$ made of metal with density p_1 g/cm^3. The second part is an inner disk with radius r_2 cm, it is made of metal with density p_2 g/cm^3. The disk is nested inside the ring.

The Olympic jury decided that r_1 will take one of possible values of $x_1, x_2, \ldots, x_n$. It is up to jury to decide which particular value r_1 will take. Similarly, the Olympic jury decided that p_1 will take one of possible value of $y_1, y_2, \ldots, y_m$, and p_2 will take a value from list $z_1, z_2, \ldots, z_k$.

According to most ancient traditions the ratio between the outer ring mass m_{out} and the inner disk mass m_{in} must equal $m_{out}/m_{in} = \frac{A}{B}$, where A, B are constants taken from ancient books. Now, to start making medals, the jury needs to take values for r_1, p_1, p_2 and calculate the suitable value of r_2.

The jury wants to choose the value that would maximize radius r_2. Help the jury find the sought value of r_2. Value r_2 doesn't have to be an integer.

Medal has a uniform thickness throughout the area, the thickness of the inner disk is the same as the thickness of the outer ring.

Input

The first input line contains an integer n and a sequence of integers $x_1, x_2, \ldots, x_n$. The second input line contains an integer m and a sequence of integers $y_1, y_2, \ldots, y_m$. The third input line contains an integer k and a sequence of integers $z_1, z_2, \ldots, z_k$. The last line contains two integers A and B.

All numbers given in the input are positive and do not exceed 5000. Each of the three sequences contains distinct numbers. The numbers in the lines are separated by spaces.

Output

Print a single real number — the sought value r_2 with absolute or relative error of at most 10^{-6}. It is guaranteed that the solution that meets the problem requirements exists.

Examples

input

```
3 1 2 3
1 2
3 3 2 1
1 2
```

output

```
2.683281573000
```

input

```
4 2 3 6 4
2 1 2
3 10 6 8
2 1
```

output

```
2.267786838055
```

Note

In the first sample the jury should choose the following values: $r_1 = 3$, $p_1 = 2$, $p_2 = 1$.

C. Crosses

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

There is a board with a grid consisting of n rows and m columns, the rows are numbered from 1 from top to bottom and the columns are numbered from 1 from left to right. In this grid we will denote the cell that lies on row number i and column number j as (i, j).

A group of six numbers (a, b, c, d, x_0, y_0), where $0 \le a, b, c, d$, is a *cross*, and there is a set of cells that are assigned to it. Cell (x, y) belongs to this set if **at least one** of two conditions are fulfilled:

- $|x_0 - x| \le a$ and $|y_0 - y| \le b$
- $|x_0 - x| \le c$ and $|y_0 - y| \le d$

The picture shows the cross $(0, 1, 1, 0, 2, 3)$ on the grid 3×4.

Your task is to find the number of different groups of six numbers, (a, b, c, d, x_0, y_0) that determine the crosses of an area equal to s, which are placed entirely on the grid. The cross is placed entirely on the grid, if any of its cells is in the range of the grid (that is for each cell (x, y) of the cross $1 \le x \le n$; $1 \le y \le m$ holds). The area of the cross is the number of cells it has.

Note that two crosses are considered distinct if the ordered groups of six numbers that denote them are distinct, even if these crosses coincide as sets of points.

Input

The input consists of a single line containing three integers n, m and s ($1 \le n, m \le 500$, $1 \le s \le n \cdot m$). The integers are separated by a space.

Output

Print a single integer — the number of distinct groups of six integers that denote crosses with area s and that are fully placed on the $n \times m$ grid.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the cin, cout streams or the `%I64d` specifier.

Examples

input

```
2 2 1
```

output

```
4
```

input

```
3 4 5
```

output

```
4
```

Note

In the first sample the sought groups of six numbers are: $(0, 0, 0, 0, 1, 1)$, $(0, 0, 0, 0, 1, 2)$, $(0, 0, 0, 0, 2, 1)$, $(0, 0, 0, 0, 2, 2)$.

In the second sample the sought groups of six numbers are: $(0, 1, 1, 0, 2, 2)$, $(0, 1, 1, 0, 2, 3)$, $(1, 0, 0, 1, 2, 2)$, $(1, 0, 0, 1, 2, 3)$.

D. Hot Days

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

The official capital and the cultural capital of Berland are connected by a single road running through n regions. Each region has a unique climate, so the i-th $(1 \le i \le n)$ region has a stable temperature of t_i degrees in summer.

This summer a group of m schoolchildren wants to get from the official capital to the cultural capital to visit museums and sights. The trip organizers transport the children between the cities in buses, but sometimes it is very hot. Specifically, if the bus is driving through the i-th region and has k schoolchildren, then the temperature inside the bus is $t_i + k$ degrees.

Of course, nobody likes it when the bus is hot. So, when the bus drives through the i-th region, if it has more than T_i degrees inside, each of the schoolchild in the bus demands compensation for the uncomfortable conditions. The compensation is as large as x_i rubles and it is charged in each region where the temperature in the bus exceeds the limit.

To save money, the organizers of the trip may arbitrarily add or remove extra buses in the beginning of the trip, and between regions (of course, they need at least one bus to pass any region). The organizers can also arbitrarily sort the children into buses, however, each of buses in the i-th region will cost the organizers $cost_i$ rubles. Please note that sorting children into buses takes no money.

Your task is to find the minimum number of rubles, which the organizers will have to spend to transport all schoolchildren.

Input

The first input line contains two integers n and m $(1 \le n \le 10^5;\ 1 \le m \le 10^6)$ — the number of regions on the way and the number of schoolchildren in the group, correspondingly. Next n lines contain four integers each: the i-th line contains t_i, T_i, x_i and $cost_i$ $(1 \le t_i, T_i, x_i, cost_i \le 10^6)$. The numbers in the lines are separated by single spaces.

Output

Print the only integer — the minimum number of roubles the organizers will have to spend to transport all schoolchildren.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use cin, cout streams or the `%I64d` specifier.

Examples

input

```
2 10
30 35 1 100
20 35 10 10
```

output

```
120
```

input

```
3 100
10 30 1000 1
5 10 1000 3
10 40 1000 100000
```

output

```
200065
```

Note

In the first sample the organizers will use only one bus to travel through the first region. However, the temperature in the bus will equal $30 + 10 = 40$ degrees and each of 10 schoolchildren will ask for compensation. Only one bus will transport the group through the second region too, but the temperature inside won't exceed the limit. Overall, the organizers will spend $100 + 10 + 10 = 120$ rubles.

E. Periodical Numbers

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

A non-empty string s is called *binary*, if it consists only of characters "0" and "1". Let's number the characters of binary string s from 1 to the string's length and let's denote the i-th character in string s as s_i.

Binary string s with length n is *periodical*, if there is an integer $1 \le k < n$ such that:

- k is a divisor of number n
- for all $1 \le i \le n - k$, the following condition fulfills: $s_i = s_{i+k}$

For example, binary strings "101010" and "11" are periodical and "10" and "10010" are not.

A positive integer x is *periodical*, if its binary representation (without leading zeroes) is a periodic string.

Your task is to calculate, how many periodic numbers are in the interval from l to r (both ends are included).

Input

The single input line contains two integers l and r ($1 \le l \le r \le 10^{18}$). The numbers are separated by a space.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the cin, cout streams or the `%I64d` specifier.

Output

Print a single integer, showing how many periodic numbers are in the interval from l to r (both ends are included).

Examples

input

```
1 10
```

output

```
3
```

input

```
25 38
```

output

```
2
```

Note

In the first sample periodic numbers are 3, 7 and 10.

In the second sample periodic numbers are 31 and 36.