A. Irrational problem

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

Little Petya was given this problem for homework:

You are given function $f(x) = (((x \mod p_1) \mod p_2) \mod p_3) \mod p_4$ (here $\mod$ represents the operation of taking the remainder). His task is to count the number of integers x in range $[a;b]$ with property $f(x) = x$.

It is a pity that Petya forgot the order in which the remainders should be taken and wrote down only 4 numbers. Each of 24 possible orders of taking the remainder has equal probability of being chosen. For example, if Petya has numbers 1, 2, 3, 4 then he can take remainders in that order or first take remainder modulo 4, then modulo 2, 3, 1. There also are 22 other permutations of these numbers that represent orders in which remainder can be taken. In this problem 4 numbers wrote down by Petya will be pairwise distinct.

Now it is impossible for Petya to complete the task given by teacher but just for fun he decided to find the number of integers $x \in [a;b]$ with property that probability that $f(x) = x$ is not less than $31.4159265352718281828459045\%$. In other words, Petya will pick up the number x if there exist at least 7 permutations of numbers p_1, p_2, p_3, p_4, for which $f(x) = x$.

Input

First line of the input will contain 6 integers, separated by spaces: p_1, p_2, p_3, p_4, a, b ($1 \le p_1, p_2, p_3, p_4 \le 1000, 0 \le a \le b \le 31415$).

It is guaranteed that numbers p_1, p_2, p_3, p_4 will be pairwise distinct.

Output

Output the number of integers in the given range that have the given property.

Sample test(s)

input
```
2 7 1 8 2 8
```
output
```
0
```

input
```
20 30 40 50 0 100
```
output
```
20
```

input
```
31 41 59 26 17 43
```
output
```
9
```

B. Energy exchange

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

It is well known that the planet suffers from the energy crisis. Little Petya doesn't like that and wants to save the world. For this purpose he needs every accumulator to contain the same amount of energy. Initially every accumulator has some amount of energy: the i-th accumulator has a_i units of energy. Energy can be transferred from one accumulator to the other. Every time x units of energy are transferred (x is not necessarily an integer) k percent of it is lost. That is, if x units were transferred from one accumulator to the other, amount of energy in the first one decreased by x units and in other increased by $x - \frac{xk}{100}$ units.

Your task is to help Petya find what maximum equal amount of energy can be stored in each accumulator after the transfers.

Input

First line of the input contains two integers n and k ($1 \le n \le 10000$, $0 \le k \le 99$) — number of accumulators and the percent of energy that is lost during transfers.

Next line contains n integers $a_1, a_2, \dots, a_n$ — amounts of energy in the first, second, .., n-th accumulator respectively ($0 \le a_i \le 1000$, $1 \le i \le n$).

Output

Output maximum possible amount of energy that can remain in each of accumulators after the transfers of energy.

The absolute or relative error in the answer should not exceed 10^{-6}.

Sample test(s)

input

```
3 50
4 2 1
```

output

```
2.000000000
```

input

```
2 90
1 11
```

output

```
1.909090909
```

C. Synchrophasotron

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

For some experiments little Petya needs a synchrophasotron. He has already got the device, all that's left is to set the fuel supply. Fuel comes through a system of nodes numbered from 1 to n and connected by pipes. Pipes go from every node with smaller number to every node with greater number. Fuel can only flow through pipes in direction from node with smaller number to node with greater number. Any amount of fuel can enter through the first node and the last node is connected directly to the synchrophasotron. It is known that every pipe has three attributes: the minimum amount of fuel that should go through it, the maximum amount of fuel that can possibly go through it and the cost of pipe activation. If c_{ij} units of fuel ($c_{ij} > 0$) flow from node i to node j, it will cost $a_{ij} + c_{ij}^2$ tugriks (a_{ij} is the cost of pipe activation), and if fuel doesn't flow through the pipe, it doesn't cost anything. Only integer number of units of fuel can flow through each pipe.

Constraints on the minimal and the maximal fuel capacity of a pipe take place **always**, not only if it is active. You may assume that the pipe is active if and only if the flow through it is strictly greater than zero.

Petya doesn't want the pipe system to be overloaded, so he wants to find the minimal amount of fuel, that, having entered the first node, can reach the synchrophasotron. Besides that he wants to impress the sponsors, so the sum of money needed to be paid for fuel to go through each pipe, must be as big as possible.

Input

First line contains integer n ($2 \le n \le 6$), which represents the number of nodes. Each of the next $n(n - 1) / 2$ lines contains five integers s, f, l, h, a that describe pipes — the first node of the pipe, the second node of the pipe, the minimum and the maximum amount of fuel that can flow through the pipe and the the activation cost, respectively. ($1 \le s < f \le n, 0 \le l \le h \le 5, 0 \le a \le 6$). It is guaranteed that for each pair of nodes with distinct numbers there will be exactly one pipe between them described in the input.

Output

Output in the first line two space-separated numbers: the minimum possible amount of fuel that can flow into the synchrophasotron, and the maximum possible sum that needs to be paid in order for that amount of fuel to reach synchrophasotron. If there is no amount of fuel that can reach synchrophasotron, output "`-1 -1`".

The amount of fuel which will flow into synchrophasotron is not neccessary positive. It could be equal to zero if the minimum constraint of every pipe is equal to zero.

Sample test(s)

input
```
2
1 2 1 2 3
```
output
```
1 4
```

input
```
3
1 2 1 2 3
1 3 0 0 0
2 3 3 4 5
```
output
```
-1 -1
```

input
```
4
1 2 0 2 1
2 3 0 2 1
1 3 0 2 6
1 4 0 0 1
2 4 0 0 0
3 4 2 3 0
```
output
```
2 15
```

input
```
3
1 2 0 2 1
1 3 1 2 1
2 3 1 2 1
```
output
```
2 6
```

Note

In the first test, we can either pass 1 or 2 units of fuel from node 1 to node 2. The minimum possible amount is 1, it costs $a_{12} + 1^2 = 4$.

In the second test, you can pass at most 2 units from node 1 to node 2, and at you have to pass at least 3 units from node 2 to node 3. It is impossible.

In the third test, the minimum possible amount is 2. You can pass each unit of fuel through two different paths: either 1->2->3->4 or 1->3->4. If you use the first path twice, it will cost $a_{12}+2^2+a_{23}+2^2+a_{34}+2^2$=14. If you use the second path twice, it will cost $a_{13}+2^2+a_{34}+2^2$=14. However, if you use each path (allowing one unit of fuel go through pipes 1->2, 2->3, 1->3, and two units go through 3->4) it will cost $a_{12}+1^2+a_{23}+1^2+a_{13}+1^2+a_{34}+2^2$=15 and it is the maximum possible cost.

Also note that since no fuel flows from node 1 to node 4, activation cost for that pipe is not added to the answer.

D. Half-decay tree

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently Petya has become keen on physics. Anna V., his teacher noticed Petya's interest and gave him a fascinating physical puzzle — a half-decay tree.

A half-decay tree is a complete binary tree with the height h. The height of a tree is the length of the path (in edges) from the root to a leaf in the tree. While studying the tree Petya can add electrons to vertices or induce random decay with synchrophasotron. Random decay is a process during which the edges of some path from the root to the random leaf of the tree are deleted. All the leaves are equiprobable. As the half-decay tree is the school property, Petya will return back the deleted edges into the tree after each decay.

After being desintegrated, the tree decomposes into connected components. Charge of each component is the total quantity of electrons placed in vertices of the component. Potential of desintegerated tree is the maximum from the charges of its connected components. Each time before inducing random decay Petya is curious about the mathematical expectation of potential of the tree after being desintegrated.

Input

First line will contain two integers h and q ($1 \le h \le 30$, $1 \le q \le 10^5$). Next q lines will contain a query of one of two types:

- `add` v e
  Petya adds e electrons to vertex number v ($1 \le v \le 2^{h+1} - 1$, $0 \le e \le 10^4$). v and e are integers.

  The vertices of the tree are numbered in the following way: the root is numbered with 1, the children of the vertex with number x are numbered with $2x$ and $2x + 1$.

- `decay`
  Petya induces tree decay.

Output

For each query `decay` solution you should output the mathematical expectation of potential of the tree after being desintegrated. The absolute or relative error in the answer should not exceed 10^{-4}.

Sample test(s)

input

```
1 4
add 1 3
add 2 10
add 3 11
decay
```

output

```
13.50000000
```

E. Contact

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Petya is preparing for the first contact with aliens. He knows that alien spaceships have shapes of non-degenerate triangles and there will be exactly 4 ships. Landing platform for a ship can be made of 3 special columns located at some points of a Cartesian plane such that these 3 points form a triangle equal to the ship with respect to rotations, translations (parallel shifts along some vector) and reflections (symmetries along the edges). The ships can overlap after the landing.

Each column can be used to land more than one ship, for example, if there are two equal ships, we don't need to build 6 columns to land both ships, 3 will be enough. Petya wants to know what minimum number of columns will be enough to land all ships.

Input

Each of 4 lines will contain 6 integers $x_1\ y_1\ x_2\ y_2\ x_3\ y_3$ $(0 \leq x_1, y_1, x_2, y_2, x_3, y_3 \leq 20)$, representing 3 points that describe the shape of each of 4 ships. It is guaranteed that 3 points in each line will represent a non-degenerate triangle.

Output

First line should contain minimum number of columns enough to land all spaceships.

Sample test(s)

input

```
0 0 1 0 1 2
0 0 0 2 2 2
0 0 3 0 1 2
0 0 3 0 2 2
```

output

```
4
```

input

```
0 0 0 1 1 1
0 0 0 2 2 2
0 0 0 5 5 5
0 0 0 17 17 17
```

output

```
9
```

Note

In the first test case columns can be put in these points: $(0, 0), (1, 0), (3, 0), (1, 2)$. Note that the second ship can land using last 3 columns.

In the second test case following points can be chosen: $(0, 0), (0, 1), (1, 0), (0, 2), (2, 0), (0, 5), (5, 0), (0, 17), (17, 0)$. It is impossible to use less than 9 columns.