TOKI Open 2014

Hosted by:

Tim Olimpiade Komputer Indonesia (TOKI)

[English: Indonesian Computing Olympiad Team]

May 8 - 10, 2014

Day	Task	Time Limit	Memory Limit
Practice	Generous Butcher	1 s	32 MB
Day 1	Social Inequality	0.3 s	64 MB
	Multiple Choice	1 s	64 MB
	Company Planning	1 s	64 MB
Day 2	Safest Route	0.2 s	128 MB
	Jump!	0.3 s	64 MB
	Fowl Sculptures	1 s	32 MB

Acknowledgment

Practice: Generous Butcher

Written by Irvan Jahja (IOI 2008). Prepared by William Gozali (IOI 2011).

Day 1: Social Inequality

Written by Derianto Kusuma (IOI 2004-2006). Prepared by Felik Junvianto.

Day 1: Multiple Choice

Written by Derianto Kusuma (IOI 2004-2006). Prepared by William Gozali (IOI 2011).

Day 1: Company Planning

Written and prepared by Jonathan Irvin Gunawan (IOI 2012-2013).

Day 2: Safest Route

Written by Derianto Kusuma (IOI 2004-2006). Prepared by William Gozali (IOI 2011).

Day 2: Jump!

Written and prepared by Ammar Fathin Sabili (IOI 2013). This was a backup task for Indonesia National Olympiad in Informatics 2013.

Day 2: Fowl Sculptures

Written by Frederikus Hudi (IOI 2011) and Ivan Wangsaciptalingga (IMO 2011). Prepared by Pusaka Kaleb Setyabudi. This was a backup task for Indonesia National Olympiad in Informatics 2013.

All tasks were proofread by Ashar Fuadi (IOI 2010) and Ilham Winata Kurnia (IOI 2002).

Special thanks to Jordan Fernando, Vincent Sebastian The, Mirza Widihananta, and the other technical team members for developing the contest environment.

Special thanks to Brian Marshal (IOI 2007) for publication of the contest.

Generous Butcher

Time Limit	1 s
Memory Limit	32 MB

Problem Description

Long time ago, there was a country that once suffered a mass hunger. There were B people who lived in the country, and each of them was starving. Luckily, a generous butcher visited the country and decided to donate A kilogram of meat.

The people decided to make sure that each person receives equal kilogram of meat. However, they faced a problem as dividing A kilogram of meat to B persons equally is not always easy. For example, if $A = 1$ and $B = 3$, then each person should receive $0.333\ldots$ kilogram of meat. Of course, it is not easy to weigh exactly $0.333\ldots$ kilogram of meat.

Finally they agreed that it was not necessary to divide the meat exactly equally. They will be satisfied if the division is accurate enough.

Before the butcher starts dividing, he wants to know the ${K_1}^{th}$ through the ${K_2}^{th}$ digits of the fractional part (in decimal) of the actual mass of meat that each person should receive. Help him.

Input Format

The first and only line contains 4 integers A, B, K_1, and K_1.

Output Format

A single line containing the ${K_1}^{th}$ through the ${K_2}^{th}$ digits of the fractional part.

Sample Input

```
3 11 2 4
```

Sample Output

```
727
```

Explanation

Each person should receive $0.272727\ldots$ kilogram of meat. The second through the fourth digits of the fractional part are 727.

Subtasks

Subtask 1 (15 points):

- $1 \leq A, B \leq 1{,}000$
- $K_1 = K_2$
- $1 \leq K_1, K_2 \leq 10$

Subtask 2 (20 points):

- $1 \leq A, B \leq 1{,}000$

- $K_1 = K_2$
- $1 \leq K_1, K_2 \leq 100{,}000$

Subtask 3 (30 points):

- $1 \leq A, B \leq 1{,}000$
- $K_1 = K_2$
- $1 \leq K_1, K_2 \leq 1{,}000{,}000{,}000$

Subtask 4 (23 points):

- $1 \leq A, B \leq 1{,}000{,}000{,}000$
- $K_1 = K_2$
- $1 \leq K_1, K_2 \leq 1{,}000{,}000{,}000$

Subtask 5 (12 points):

- $1 \leq A, B \leq 1{,}000{,}000{,}000$
- $0 \leq K_2 - K_1 \leq 100{,}000$
- $1 \leq K_1, K_2 \leq 1{,}000{,}000{,}000$

Social Inequality

Time Limit	0.3 s
Memory Limit	64 MB

Problem Description

Dream Nation's Department of Public Welfare recently launched a plan to improve the social welfare of Dream Nation's citizens. Before they can develop the appropriate strategies and programs, they need to know which cities are suffering the social inequality problem and how critical this problem is.

An employee of the department proposes a method for measuring the level of social inequality in a city. Suppose we know that a particular city has N citizens, numbered 1 through N. The i^{th} citizen lives at location L_i and has income P_i. We define the level of personal inequality between the i^{th} citizen and the j^{th} citizen as the product of absolute difference of the locations and the absolute difference of incomes. In other words, the level of personal inequality $= |L_i - L_j| \times |P_i - P_j|$. The level of social inequality of a city is the sum of the personal inequality levels of each distinct pair of its citizens.

For example, suppose we have a city with the following data of citizens:

Number	Location	Income
1	5	10
2	5	10
3	5	14
4	3	11

- The personal inequality level between the 1^{st} and 2^{nd} citizen is $|5-5| \times |10-10| = 0$.
- The personal inequality level between the 1^{st} and 3^{rd} citizen is $|5-5| \times |10-14| = 0$.
- The personal inequality level between the 1^{st} and 4^{th} citizen is $|5-3| \times |10-11| = 2$.
- The personal inequality level between the 2^{nd} and 3^{rd} citizen is $|5-5| \times |10-14| = 0$.
- The personal inequality level between the 2^{nd} and 4^{th} citizen is $|5-3| \times |10-11| = 2$.
- The personal inequality level between the 3^{rd} and 4^{th} citizen is $|5-3| \times |14-11| = 6$.

Therefore, the social inequality level of the city is $2 + 2 + 6 = 10$.

Given the data of citizens of the city, help the Department of Public Welfare calculate the social inequality level of the city.

Input Format

The first line consists of a single integer N. Each of the following N lines contains 2 integers L_i and P_i.

Output Format

A single line containing the social inequality level of the given city. The answer is guaranteed to fit into 64-bit integer data type.

Sample Input

```
4
5 10
5 10
5 14
3 11
```

Sample Output

```
10
```

Subtasks

Subtask 1 (29 points):

- $1 \le N \le 5{,}000$
- $1 \le L_i \le 100$
- $1 \le P_i \le 100$

Subtask 2 (19 points):

- $1 \le N \le 50{,}000$
- $1 \le L_i \le 100$
- $1 \le P_i \le 100$

Subtask 3 (20 points):

- $1 \le N \le 50{,}000$
- $1 \le L_i \le 100$
- $1 \le P_i \le 10{,}000$

Subtask 4 (32 points):

- $1 \le N \le 100{,}000$
- $1 \le L_i \le 10{,}000$
- $1 \le P_i \le 10{,}000$

Multiple Choice

Time Limit	1 s
Memory Limit	64 MB

Problem Description

A good teacher facilitates the students to learn through various ways including the exams. For example, a teacher could provide feedback to the students by providing them with a/the correct answer key to the exam questions. However, some teachers have little interest in doing so.

Anto just had a Bahasa Indonesia exam. The exam consists of N multiple choice problems. Each problem has 5 choices for the answer, labeled 'a', 'b', 'c', 'd', and 'e'. Out of the N problems, Anto answered P problems correctly. Anto is curious to know the correct answer to each problem. However, his teacher never tells him.

Anto has an idea. He borrowed the exam result of his friend, Budi, who answered Q problems out of the N problems correctly. For the i^{th} problem, Anto answered A_i and Budi answered B_i. A_i and B_i are elements of {'a', 'b', 'c', 'd', 'e'}. Additionaly, Anto knows the "must-be-wrong" choices. For example, for problem 1, he surely knows that the correct answer is one of 'a', 'c', or'd', and for problem 2, the correct answer is either 'a' or 'e', etc.

With this information, Anto hopes that he can figure out the correct answer to each problem. Unfortunately, there might be more than one answer key that is consistent with the information. Help Anto compute the number of possible answer keys.

Input Format

The first line contains 3 integers N, P, and Q. The next line contains N characters $A_1, A_2, A_3, \ldots, A_N$ stating Anto's answers. The next line contains N characters $B_1, B_2, B_3, \ldots, B_N$ stating Budi's answers. Each answer is separated by a single space.

The next N lines are choices for the correct answer, from the first problem to the N^{th} problem. Each line contains 5 characters:

- First character will be either 'a' or '.'
- Second character will be either 'b' or '.'
- Third character will be either 'c' or '.'
- Fourth character will be either 'd' or '.'
- Fifth character will be either 'e' or '.'

A character from {'a', 'b', 'c', 'd', 'e'} exists on the i^{th} line <u>if and only if</u> that choice is a possible answer for the i^{th} problem. For each problem i, it is guaranteed that characters A_i and B_i are present on that line. For example, a line containing "a.cd." means that Anto knows that the answer for that problem is either 'a', 'c', or 'd'.

Output Format

A single line containing the number of answer keys that satisfy the given information, modulo 1,000,000,007. It is possible that no answer keys exist (the teacher might have made a mistake!).

Sample Input 1

```
5 4 3
a b c d c
a b d d c
a.cd.
.b...
..cde
...d.
..cd.
```

Sample Output 1

```
3
```

Sample Input 2

```
6 3 3
a b c d c d
a b d c d c
abc..
ab...
..cde
..cde
..cde
..cde
```

Sample Output 2

```
30
```

Explanation

For the first sample, there are 3 possible answer keys such that Anto correctly answered 4 problems and Budi correctly answered 3 problems:

- a b c d d
- c b c d c
- d b c d c

For the second sample, the possibilities are:

- a b c c e e
- a b d d e e
- a b c e d e
- a b d e c e
- ... (there are 12 answer keys starting with a b)
- a a c c c c
- a a c c d d
- a a d d c c
- a a d d d d
- a a c d d c
- a a d c c d
- b b ... (similar, there are 6)

- c b ... (similar, there are 6)

Subtasks

For each subtask,

- For each problem i, it is guaranteed that characters A_i and B_i exist on the $(3+i)^{th}$ line of the input.

Subtask 1 (24 points):

- $1 \leq N \leq 200$

Subtask 2 (16 points):

- $1 \leq N \leq 2{,}000$
- $A_i \neq B_i$ for each i.
- For each problem i, Anto knows that there are exactly 2 possible correct answer choices.

Subtask 3 (60 points):

- $1 \leq N \leq 2{,}000$

Company Planning

Time Limit	1 s
Memory Limit	64 MB

Problem Description

There are N employees in a company, numbered 1 through N. The boss of the company is numbered 1. Each employee except the boss has exactly one supervisor. For each employee numbered x (except the boss), his/her supervisor number is y, where $1 \leq y < x$. Employee x is then called a subordinate of employee y. This relationship is not transitive: employee x is not called a subordinate of the supervisor of employee y. Each employee may have zero or more subordinates.

The company faces a restructuring program called "Company Planning". Through this program each employee in the company is not allowed to have more than K subordinates. If an employee has more than K subordinates, he/she has to fire some of his/her subordinates. Furthermore, if an employee is fired, then all of his/her subordinates will also be fired. This rule is applied recursively: all of his/her subordinates' subordinates will be also fired, and so on.

The company would like to retain as many employees as possible. For that reason, the company wants to know the minimum possible value of K, such that the company can still have at least M employees. Help the company.

Input Format

The first line contains 2 integers N and M. This will be followed by $N - 1$ lines. The ith of these lines contains the number of supervisor of employee number $(i + 1)$.

Output Format

A single integer containing the minimum possible value of K.

Sample Input

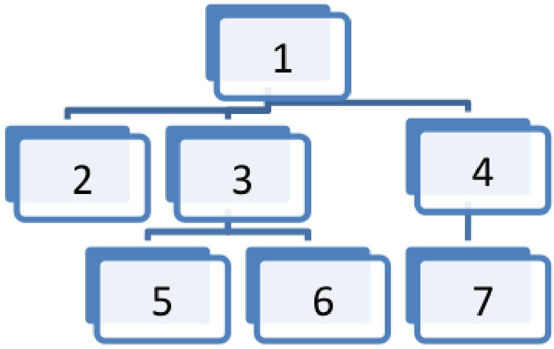
```
7 5
1
1
1
3
3
4
```

Sample Output

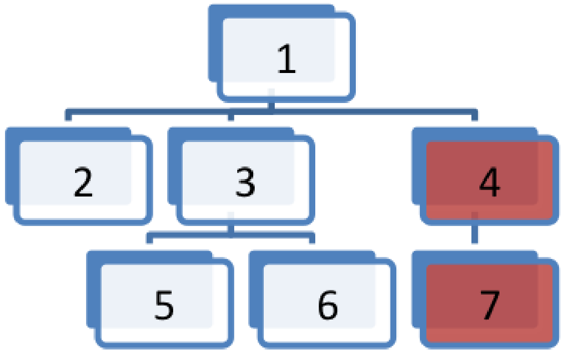
```
2
```

Explanation

This is an illustration of the sample input:



By firing employee 4, the company will have 5 remaining employees. Each of these employees does not have more than 2 subordinates. This is the minimum possible value of K.



Subtasks

Subtask 1 (12 points):

- $1 \le M \le N \le 10$

Subtask 2 (20 points):

- $1 \le M \le N \le 1{,}000$

Subtask 3 (18 points):

- $1 \le M \le N \le 100{,}000$
- Each employee will have at most 2 subordinates.

Subtask 4 (7 points):

- $1 \le N \le 100{,}000$
- $M = N$

Subtask 5 (43 points):

- $1 \le M \le N \le 100{,}000$

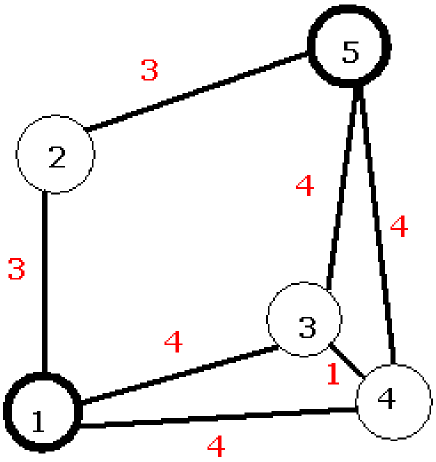
Safest Route

Time Limit	0.2 s
Memory Limit	128 MB

Problem Description

Mr. Dengklek loves traveling, and he often asks programmers to help him find the shortest path from a city to another city. After years of traveling, he realized that shortest path is not always the most time-efficient path. This can happen when some roads in the shortest path is in such poor condition causing the need to turn around and find another path. So this time, he is interested in finding the safest route.

There are V cities and E roads. The cities are numbered 1 through V. Each road connects two distinct cities and is bidirectional (two-way). For example, please take a look at this picture.



Each circle denotes a city and its number, while each line denotes a road and its length.

If Mr. Dengklek wants to travel from city 1 to city 5, the shortest path can be achieved by going to city 2, then city 5. This route is written as <1, 2, 5>. However, Mr. Dengklek needs to know the shortest path for a not-so-ideal condition. For example, if a road from city 1 to city 2 is damaged, the shortest path will become <1, 3, 5> whose total distance is 8. If the road from city 1 to city 2 is OK, but the road from city 2 to city 5 is damaged, then Mr. Dengklek needs to go back from city 2 back to city 1, then city 3, and finally city 5. The total distance traveled in this scenario is $3 + 3 + 8 = 14$.

To travel from city X_1 to city X_N with route <X_1, X_2, X_3, ..., X_N>, Mr. Dengklek defines not-so-ideal length for that route as:

```
max {
  distance(X1, X2, X3, ..., XN) assuming no roads are damaged

  shortest path from X1 to XN assuming the road from city X1 to
  city X2 is damaged,

  distance(X1, X2) + shortest path from X2 to XN assuming the
  road from city X2 to city X3 is damaged,

  distance(X1, X2, X3) + shortest path from X3 to XN assuming the
```

```
  road from city X₃ to city X₄ is damaged,

  ...,

  distance(X₁, X₂, X₃, ..., X_{N-2}) + shortest path from X_{N-2} to X_N
  assuming the road from city X_{N-2} to city X_{N-1} is damaged,

  distance(X₁, X₂, X₃, ..., X_{N-1}) + shortest path from X_{N-1} to X_N
  assuming the road from city X_{N-1} to city X_N is damaged
}
```

where distance(X_1, X_2, X_3, ..., X_P) denotes the total distance traveled from X_1 to X_2, then from X_2 to X_3, ..., until from X_{P-1} to X_P.

According to that definition, route <1, 2, 5> has not-so-ideal length $= max(6, 8, 14) = 14$. There exists a better route, which is <1, 3, 5>. If there are no damaged roads, the shortest path is 8. If the road from city 1 to city 3 is damaged, the shortest path is 6 by visiting city 2 and then city 5. If the road from city 3 to city 5 is damaged, after traveling from city 1 to city 3, Mr. Dengklek can go to city 4, and finally city 5, having total distance of 9. So, the not-so-ideal length for route <1, 3, 5> is $max(8, 6, 9) = 9$.

Finally, Mr. Dengklek defines a safest route as a route which has the smallest not-so-ideal length, and its value as the not-so-ideal length for that route. So, for the example above a safest route is <1, 3, 5> and its value is 9.

You are given the cities and roads information, starting city (S), and destination city (T). Find the value of a safest route.

Input Format

The first line contains 4 integers, V, E, S, and T. Each of the next E lines contains 3 integers a, b, and c denoting a bidirectional road connecting city a and city b, with length c.

Output Format

A single line containing the value of a safest route.

Sample Input 1

```
5 7 1 5
1 2 3
2 5 3
1 3 4
3 5 4
1 4 4
4 5 4
4 3 1
```

Sample Output 1

```
9
```

Sample Input 2

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

Sample Output 2

```
3
```

Explanation

The first sample is the example explained in the problem description.

For the second sample, the only safest route is <1, 2, 4>, with not-so-ideal length $= max(3, 3, 3) = 3$. For comparison, route <1, 3, 4> has not-so-ideal length = 5, and route <1, 2, 3, 4> has not-so-ideal length = 5.

Subtasks

For each subtask,

- $1 \leq S, T \leq V$
- $S \neq T$
- $1 \leq E \leq V \times (V - 1)/2$
- $1 \leq a, b \leq V$
- $1 \leq c \leq 10{,}000$
- Each road connects two distinct cities.
- For each pair of cities, at most one road connects them.
- If a road is damaged, it is guaranteed that there exists at least one alternative path from S to T.

Subtask 1 (14 points):

- $1 \leq V \leq 6$

Subtask 2 (30 points):

- $1 \leq V \leq 100$

Subtask 3 (25 points):

- $1 \leq V \leq 250$

Subtask 4 (18 points):

- $1 \leq V \leq 500$

Subtask 5 (13 points):

- $1 \leq V \leq 2{,}500$
- Initially, there are exactly two simple paths from S to T.

Jump!

Time Limit	0.3 s
Memory Limit	64 MB

Problem Description

Mr. Dengklek has a smart frog called Si Katak. Si Katak lives in a circular pond. Mr. Dengklek has an unlimited number of stones, numbered 1, 2, 3, ... Initially, he surrounds the pond clockwise with stones 1, 2, ..., N in that order. Mr. Dengklek also assigns scores to the stones such that stone i is assigned A_i points. The other stones have 0 points.

Mr. Dengklek loves to request Si Katak to jump on the stones. For each request, Si Katak starts on stone 1 and picks a non-negative integer P. If $P = 0$, then it decides to ignore Mr. Dengklek's request. Otherwise the following happens. Si Katak jumps clockwise to the next stone repeatedly. As the pond is circular, stone N's next stone is stone 1. After P jumps, Si Katak stops on a stone. Mr. Dengklek marks that stone and then Si Katak continues to jump until it jumps P times, and so on. Si Katak stops its actions right after stone 1 has been marked.

Deliberately, Si Katak will never pick $P = 1$. It will also never pick $P \geq N - 1$ if $N \geq 2$. This is because the total distance of its jumps will be too near or too far. So, P will always be between 2 and $N - 2$, inclusive, or 0. If $P = 0$, then Mr. Dengklek will only mark stone 1.

Finally, Mr. Dengklek will accumulate the scores on the marked stones. Si Katak has to pick P in such a way that the total score is maximum.

To be more challenging, sometimes Mr. Dengklek changes the value of N, the number of stones surrounding the pond. Suppose that Mr. Dengklek changes the value of N to M. Then,

- If $M < N$, then Mr. Dengklek will remove stones $M + 1$, $M + 2$, ..., N. The removed stones will retain their scores.
- If $M > N$, then Mr. Dengklek will add stones $N + 1$, $N + 2$, ..., M clockwise in that order between stone N and stone 1.

After that, Mr. Dengklek sets $N = M$.

Not only that, sometimes Mr. Dengklek changes the score of a stone. Mr. Dengklek can only change the score of stones that are currently surrounding the pond (stones 1, 2, ..., N).

Help Si Katak determine the value of P for each of Mr. Dengklek's jump requests, such that the total score of the resulting marked stones is maximum.

Input Format

The first line contains 2 integers N and Q. N is the number of stones surrounding the pond initially, and Q is the number of operations. The next line contains N integers $A_1 \ldots A_N$.

Each of the next Q lines contains one of these actions:

- `JUMP`, Si Katak must jump in such a way that the total score of marked stones is maximum.
- `RESIZE M`, Mr. Dengklek changes the value of N to M.

- `UPDATE X Y`, Mr. Dengklek changes the score of stone X into Y.

Output Format

For each `JUMP` request, output the maximum possible total score of Si Katak's jumps.

Sample Input

```
8 8
3 -1 -100 0 8 1 5 2
JUMP
RESIZE 4
JUMP
RESIZE 8
JUMP
UPDATE 3 4
RESIZE 9
JUMP
```

Sample Output

```
11
3
11
22
```

Explanation

For the first request, Si Katak should pick $P = 4$. The marked stones are stone 5 and stone 1 in that order. The total score will be $A_5 + A_1 = 11$.

Next, the value of N is changed to 4. So, stones 5, 6, 7, 8 are removed. The only possible values of P are 0 and 2. The highest possible total score of Si Katak's jumps will be 3 (when $P = 0$).

Next, the value of N is changed to 8. So, stones 5, 6, 7, 8 are added back. Hence, Si Katak should pick $P = 4$ again and will score 11 points.

Finally, the score of stone 3 is changed from -100 points to 4 points, and the value of N is changed to 9. The score of stone 9 is 0. Si Katak could choose $P = 4$. The marked stones will be stones 5, 9, 4, 8, 3, 7, 2, 6, 1 in that order. The total score is 22.

Subtasks

For each subtask,

- $1 \leq i \leq N$
- $1 \leq X \leq N$
- $-1{,}000{,}000 \leq Y \leq 1{,}000{,}000$
- $-1{,}000{,}000 \leq A_i \leq 1{,}000{,}000$

Subtask 1 (13 points):

- $1 \leq N \leq 100$ at any moment.
- $1 \leq Q \leq 100$

Subtask 2 (27 points):

- $1 \le N \le 2{,}000$ at any moment.
- $1 \le Q \le 1{,}000$

Subtask 3 (24 points):

- $1 \le N \le 2{,}000$ at any moment.
- $1 \le Q \le 50{,}000$
- There will be no `RESIZE` actions.

Subtask 4 (36 points):

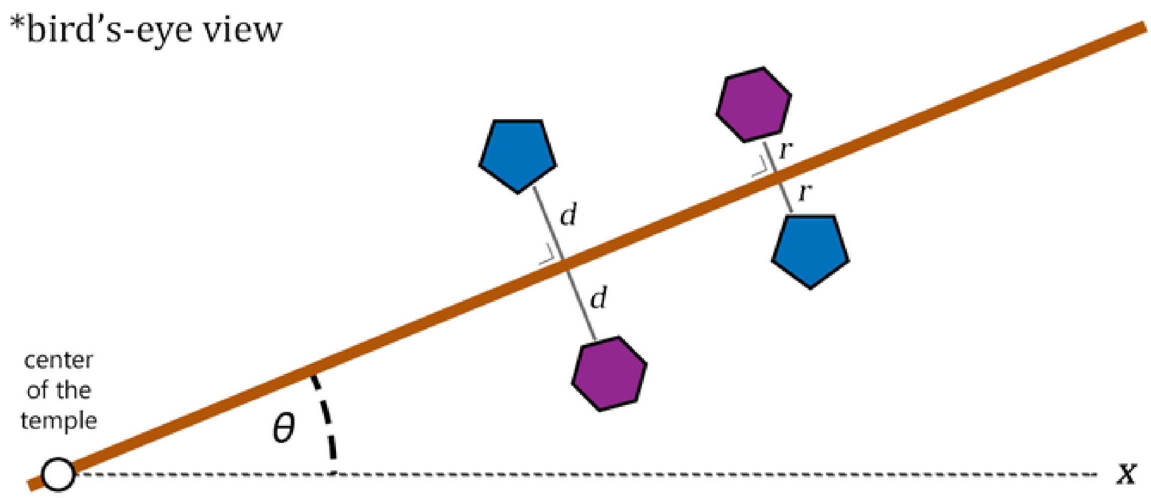
- $1 \le N \le 2{,}000$ at any moment.
- $1 \le Q \le 50{,}000$

Fowl Sculptures

Time Limit	1 s
Memory Limit	32 MB

Problem Description

Mr. Ganesh manages the conservation of the biggest temple in the world: the TOKI Temple. From old records, Mr. Ganesh concludes that the temple has two interesting objects: the central point (center of the temple) and the main path. The main path of TOKI Temple is a straight line that passes the central point. For convenience, we represent the temple setting in the Cartesian coordinate system. The central point is located at the origin $(0, 0)$, and the main path is a line forming the angle θ with the positive X axis (counterclockwise).



There are two kinds of sculptures in TOKI Temple: chicken sculptures and goose sculptures. The position of each chicken sculpture is always symmetric to another goose sculpture with respect to the main path of the TOKI Temple. Unfortunately, because of the accumulation of dust the main path of the TOKI Temple is no longer visible. This means nobody knows the exact value of θ.

The number of sculptures inside the temple is still unknown. During the conservation, Mr. Ganesh has found N sculptures. None of them is located at $(0, 0)$. Each sculpture is found in a poor condition such that it is indistinguishable whether it is a chicken or a goose sculpture. In order to identify the type of each sculpture, Mr. Ganesh just made a public contest to find other missing sculptures (if any).

Mr. Dengklek, who soon heard about the contest, is eager to win it. He realized that there must be at least K missing sculptures besides the N sculptures that have been found, thanks to the symmetry property.

What is the minimum possible value of K, and what is a possible value of θ for this K?

Input Format

The first line of contains an integer N. Each of the next N lines contains 2 integers x_i and y_i, the coordinates of the i^{th} sculpture.

Output Format

The first line should contain a floating-point number θ $(0 \leq \theta < 180)$ that denotes the slope of the main path in degrees, rounded to 3 decimal places.

The second line should contain an integer K denoting the minimum number of missing sculptures if the angle is θ.

If there are several values of θ that yield the same value of K, choose the largest value of θ.

Sample Input 1

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

Sample Output 1

```
0.000
1
```

Sample Input 2

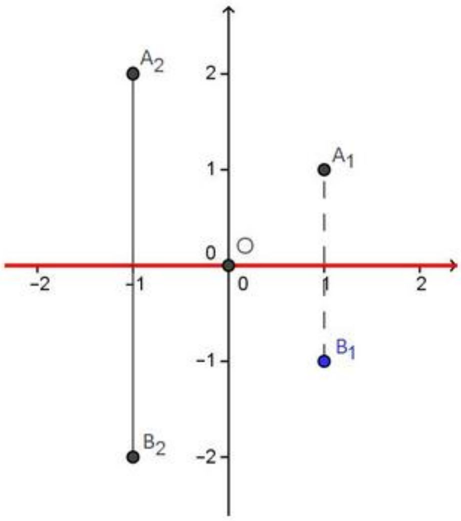
```
4
-1 1
1 0
0 1
1 -1
```

Sample Output 2

```
45.000
0
```

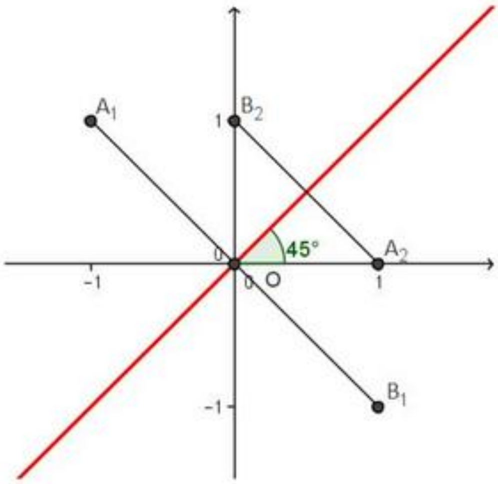
Explanation

This is the illustration of sample input 1:



Here, A_i denotes the i^{th} chicken sculpture, and B_j denotes the j^{th} goose sculpture. The blue dots represent the postulated locations of the undiscovered sculptures, while the black dots represent the location of the sculptures that have already been found. The red line represents the main path of TOKI Temple.

This is the illustration of sample input 2:



Subtasks

For each subtask,

- $-1{,}000{,}000{,}000 \le x_i, y_i \le 1{,}000{,}000{,}000$
- No sculptures (both missing and already found) are located on $(0,0)$.
- No two or more sculptures are located on the same coordinate.

Subtask 1 (17 points):

- $1 \le N \le 200$
- It is known that $\theta = 0$.

Subtask 2 (23 points):

- $1 \le N \le 200$

- It is known that $0 \leq \theta < 180$.

Subtask 3 (60 points):

- $1 \leq N \leq 2{,}000$
- It is known that $0 \leq \theta < 180$.