IDI Open Programming Contest March 21st, 2021

Problem Set

Jury and Problem Writers

Nils Barlaug
Sander Land
Dmitry Ivankov
Sondre Sortland
Edvard Kristoffer Karlsen
Jean Niklas L'orange (Head Judge)

Tips

- Tear the problem set apart and share the problems among you.
- Problems are not ordered by difficulty.
- Try solving the easy problems first. Two problems in this set are tagged with "(Easy)" to help point you in the right direction.
- If your solution fails on a problem, you can print your program and debug it on paper while you let someone else work on a different problem on the computer.
- If you need help, contact the judges.
- Look at the scoreboard if you are unsure which problem to work on next.

Rules

- Each team consists of one to three contestants.
- One computer is used per team.
- You may not cooperate with persons not on your team.
- You may print your programs on paper to debug them.
- What you may bring to the contest:
  - Any written material (Books, manuals, handwritten notes, printed notes, etc).
  - Pens, pencils, blank paper, stapler and other useful non-electronic office equipment.
  - NO material in electronic form (CDs, USB pen and so on).
  - NO electronic devices (PDAs and so on).
- The only electronic content you may consult during the content is that specified by the organiser (see the web-page). You may not copy source code from web pages, etc.
- Your programs should read from standard in and write to standard out. Writing to standard error will result in a failed submission. C programs should return 0 from `main()`.
- Your programs may not:
  - access the network,
  - read or write files on the system,
  - talk to other processes,
  - fork,
  - or similar stuff.
  - If you try, your program will hang or crash. If it hangs, it will take a couple of minutes before others will be able to run their programs. So please make an effort to not crack/break what we have spent our spare time preparing for you.
- Show common sense and good sportsmanship.

Asteroid Mining

Problem ID: asteroidmining

Shofa is working on her new spacefaring game, *No Person's Star*. The star systems in the game are procedurally generated, and contain stars, planets and asteroids. The asteroids are only used to mine minerals, and consist of multiple mineral veins. Players can only mine one vein at a time, and they will always mine the vein with the most minerals first.

One of the clever things Shofa has done is to base asteroids on two previously generated asteroids. To generate a new asteroid i, she copies all mineral veins from asteroid a_{i1} to the new asteroid, and scales them all up by multiplying with the factor f_{i1}. She then does it again with asteroid a_{i2} and the factor f_{i2}.

For example, if asteroid 0 contains the mineral veins $[3.0, 4.0, 5.0]$, and asteroid 1 contains the mineral veins $[14.0, 56.0]$, then combining them with $f_{i1} = 1.1$ and $f_{i2} = 0.5$ will create a new asteroid with the mineral veins $[3.3, 4.4, 5.5, 7.0, 28.0]$.

This can be a bit iffy, because newer asteroids could grow their total mineral amount exponentially! However, she thinks this will not be a problem in practice if she generates them based on the current state of the parent asteroids: Hopefully people will stay at the same asteroid until it's out of resources.

To test this hypothesis, she has gotten some prototype testers to play the game for her, and she is now parsing the logs. Could you help her double check the results for her?

Input

The first line contains three integers, A, A_{init} and E: The total number of asteroids at the end, the number of initial asteroids, and the number of events in the event log.

Next follow $2A_{init}$ lines describing the contents of the initial asteroids, two lines per asteroid:

- The first line contains an integer n_i, the number of mineral veins in this asteroid.
- The second line contains n_i real numbers $m_{i,j}$, each describing the amount of minerals in a single vein.

Then follow E lines, one per event, ordered by their event time:

- `mine` a_i g_i – denoting a player mining the g_i most mineral-rich veins on asteroid a_i
- `generate` a_{i1} f_{i1} a_{i2} f_{i2} – denoting the creation of the next asteroid, generated as explained above

The first `generate` event creates the asteroid with index A_{init}, the second with index $A_{init} + 1$ and so on.

Output

For each asteroid, print a line containing the total amount of minerals in the asteroid along with the total amount of minerals in the biggest mineral vein (or 0 if there are no more veins left). The numbers must have an absolute or relative error of at most 10^{-6}, and can be printed in scientific notation.

Limits

- $0 < A \leq 350$
- $0 < A_{init} \leq \min(10, A)$
- $0 < E \leq 700$
- $0 < n_i \leq 1\,000$ and $0.0 < m_{i,j} < 200.0$
- Asteroids have been created before they are first mentioned in the event log
- Asteroids will always contain less than 10^{222} minerals in total
- For `mine` events: $0 < g_i \leq size(a_i)$ and $g_i \leq 300$, where $size(a_i)$ is the current number of mineral veins on asteroid a_i

- For `generate` events: $0.5 \leq f_{i1}, f_{i2} \leq 2.0$
- Real numbers in the input will have at most 4 digits after the decimal point

Sample Input 1

```
4 2 5
3
3.0 4.0 5.0
4
14.0 56.0 57.0 105.0
mine 1 2
generate 0 1.1 1 0.5
mine 1 1
generate 2 1.1 1 2.0
mine 3 2
```

Sample Output 1

```
12.0 5.0
14.0 14.0
48.2 28.0
22.22 7.7
```

Bootstrapping Number

Problem ID: bootstrappingnumber

Neelix has been working on making the Altonian warp bubble power itself up. While waiting for the system checks to complete, he saw a number on his screen and wondered if there is a number that powers itself up to this number.

Input

The first and only input line consists of a single integer n, the number Neelix saw on his screen.

Output

Output the number x, where $x^x = n$. Your answer must have an absolute or relative error of at most 10^{-6}.

Limits

- $1 \leq n \leq 10\,000\,000\,000$

Sample Input 1	Sample Output 1
4	2.0

Sample Input 2	Sample Output 2
42	3.20720196137

Captured by Aliens

Problem ID: capturedbyaliens

"While the Baroque rules of Chess could only have been created by humans, the rules of Go are so elegant, organic, and rigorously logical that if intelligent life forms exist elsewhere in the universe they almost certainly play Go." – Edward Lasker

Astounding news, it was found that aliens exist, and they do play Go! To check that no planets with intelligent life are in the way of an intergalactic highway construction project for a hyperspace express route, they have broadcast a number of game records to see if there is any intelligent response. It is up to you to reply to the aliens with the number of stones captured in each game, and prevent the imminent destruction of our planet.

In the game of Go, there are two players (black and white) who alternate moves, placing a single stone of their colour on an empty intersection of a square board with $S \times S$ intersections. Two intersections are *adjacent* if they are connected by a horizontal or vertical line with no other intersections between them. A group of stones is defined by taking a single stone and adding all other stones of the same colour which can be reached through repeatedly following adjacent stones. A group is captured if it is surrounded by the opponent such that it has no empty adjacent intersections. Note that spaces beyond the edges of the board do not count as empty. A captured group is removed from the board, the intersections it used to occupy become empty, and can be played on again. If a move would cause a group from both players to be captured, only the group of the player who didn't make the last move gets captured. Note that it is possible to capture multiple groups with the same move.

NB. unlike in most human rule sets, there are no further restrictions: the aliens are fine with 'suicidal' moves that cause a player's own stones to get captured, or a *ko* pattern in which stones repeatedly capture each other back and forth. In addition, due to their brains being the size of the universe, they prefer arbitrarily sized boards over our usual 19×19 sized boards.

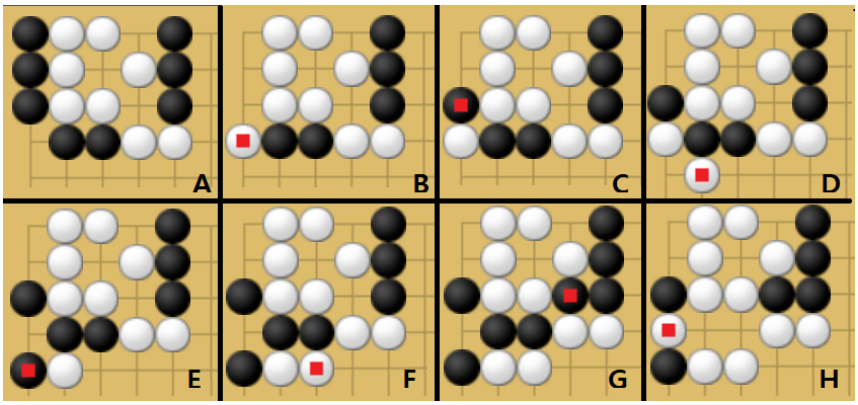


Figure 1: Example of the top left of a board with moves and captures. The position (A) starts with three black groups and three white groups. Each move is marked in red. In (B), white captures three black stones. In (E) black captures one white stone. In (H) white captures two black stones. In total, six stones were captured.

Input

The first line contains two integers, S, M: the size of the board, and the total number of moves played. The board has $S \times S$ intersections.

Next follow M lines describing the moves, each with two integers $0 \leq x, y < S$, the coordinates of the move played.

Output

Output a single line with the total number of captured stones.

Limits

- $0 < S \leq 10^5$
- $0 < M \leq 25\,000$

Sample Input 1	Sample Output 1
2 6 0 1 0 0 1 0 0 0 0 0 1 1	5

Damaged Equation

Problem ID: damagedequation

The scientists Mildra and Gloisa have been responsible for exploring Luyten 726-8, but have ended up with an issue: Their fusion reactor has stopped working! In order to continue their work, they need to get some materials from the closest asteroid. To do that though, they need to start up the chemical engines, which haven't been started for multiple centuries.

Gloisa has been reading the manual for the last three weeks, and is almost able to start it up. However, there is one equation she needs that has some operators smudged out. That equation is

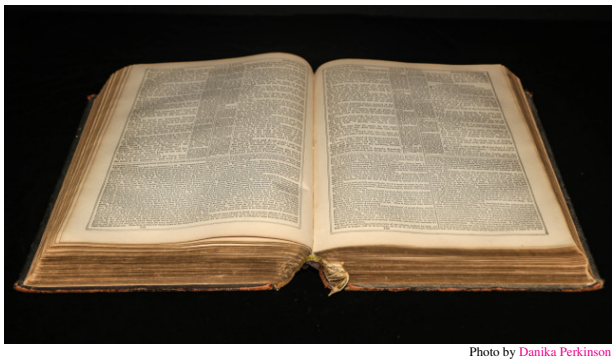


Photo by Danika Perkinson

$$a \ ? \ b \ = \ c \ ? \ d$$

where the question marks are the unknown operators. So far, Gloisa has only managed to narrow the operators down to $+$, $-$, $*$ and $/$ (integer division).

Although there may be multiple valid sets of operators for this equation, Mildra wants to help out by making a program that narrows the possibility space for Gloisa.

Input

The first line and only line consist of 4 integers, a, b, c and d, the values in the expression above.

Output

Print out all the expressions that evaluate to true, separated by a newline and sorted alphanumerically. (`'*' < '+' < '-' < '/'`)

A valid expression must have a space between the integers, operators and the equal sign. Note that expression including division by zero is not a valid expression.

If none of the expressions are valid, print out "`problems ahead`" (without the quotes).

Limits

- $0 \leq a, b, c, d \leq 100$

Sample Input 1

```
1 2 3 4
```

Sample Output 1

```
1 - 2 = 3 - 4
1 / 2 = 3 / 4
```

Sample Input 2

```
0 4 2 2
```

Sample Output 2

```
0 * 4 = 2 - 2
0 + 4 = 2 * 2
0 + 4 = 2 + 2
0 / 4 = 2 - 2
```

Sample Input 3

```
4 5 9 8
```

Sample Output 3

```
problems ahead
```

Eternal Embers
Problem ID: eternalembers

Chad was a firefighter, a very good one at that. Not only was he good at extinguishing fires, he also knew the city's design by heart. This made him able to find the fastest route from the fire station to anywhere else in the city quite quickly, saving a lot of lives.

But Chad was mean-spirited when he was not firefighting. He frequently stole bus seats from old ladies, vacuumed his apartment at 6 AM on Sundays, and replaced all the sugar at work with salt.

So it was a tough choice for the Maker when They had to decide whether Chad should get to go Upstairs or Downstairs. Clearly, the lives he had saved outweighed his mean-spiritedness, but it would not be good for the community Upstairs if Chad became a member. The Maker decided therefore that Chad should go Downstairs. Whether this is morally the right choice or not is outside of the scope of this problem.

As the worst thing Chad did was being mean to other people, the rulers of the Downstairs decided to let Chad get off easy: They gave Chad a fire hose with infinite capacity, set up some pits with eternal flames, and then tasked him to extinguish the flames. But the fires would only turn into eternal embers, which would turn back into eternal flames again after exactly A seconds! If he weren't able keep them all extinguished A seconds after his work shift started until the work shift ended, he would receive severe punishment: No shower that day.

Chad is still good at extinguishing fires, using no time at all to do so. However, he was never told A, has no knowledge of the Downstair's layout, and tended to wander all over the place. After some weeks, the unfortunate canteen imps couldn't handle his odour anymore and gave Chad some hints on how he would be able to shower. They gave him a map over the Downstairs, and although they didn't know what A is, they knew how it was computed: A is exactly the time it takes to visit all the flames in the shortest amount of time, and return back to the main office.

Chad always starts the workday by extinguishing the eternal flames right outside the main office. Can you help Chad find the length of the optimal route, so that the poor imps can regain a health and safety-compliant work environment?

Input

The first line contains three integers, N, $|V|$ and $|E|$, denoting the number of eternal flame pits, the number of locations, and the number of paths between the locations. The flame pits are situated at locations numbered $0 \ldots N-1$, main office is situated at location 0.

Then follow $|E|$ lines, each representing a path between two locations. Each line contains three integers, u_i, v_i and w_i, denoting the two locations and the time it takes to walk from one location to the other in seconds. It is possible to walk both from u_i to v_i and vice versa.

Output

Output A, the length of the optimal route as described above.

Limits

- $0 < N \leq 12$
- $N \leq |V| \leq 300$
- $|V| - 1 \leq |E| \leq |V|^2$
- $0 \leq u_i < v_i < |V|$
- $0 < w_i \leq 1\,000$
- $G = (V, E)$ forms a fully-connected undirected graph.

Sample Input 1	Sample Output 1
3 5 5 0 4 6 1 4 7 1 3 6 2 3 7 2 4 5	36

Fake News!
Problem ID: fakenews

Excitement is rapidly increasing in anticipation of the concluding debate at the 0x7E4 Undemocratic Inclinational Unconvention, where the party's candidate for the office of Student Assistant to the Old Chemistry Building Assistant Printer Service Technician's Assistant will be elected. To secure a revered lifetime appointment as antenna calibrator for the `dank.nt.nu` pirate radio station (broadcasted live from the Gløshaugen catacombs every night at 00:13:37 CET), you have been challenged to produce a 256-minute feature revealing the *character type* of each candidate.

It is well known any candidate is either a *truther*, who always tells the truth, a *fabulist*, who never tells the truth, or a *charlatan*, who starts any debate speaking truthfully, but eventually switches to uttering only falsehoods. (Precisely, a charlatan first utters one or more true statements, then she utters one or more false statements, and that is all.)

Now, as should be expected candidates talk nothing about printer maintenance policy, but about only each other's character types. In particular, candidates utter propositions using the following language:

- `truther <n>`, where `<n>` is a name of a candidate, which is true if `<n>` is the name of a truther
- `fabulist <n>`, where `<n>` is a name of a candidate, which is true if `<n>` is the name of a fabulist
- `charlatan <n>`, where `<n>` is a name of a candidate, which is true if `<n>` is the name of a charlatan
- `not <p>`, where `<p>` is a proposition, which is true if `<p>` is false
- `and <p> <q>`, where `<p>` and `<q>` are propositions, which is true if both `<p>` and `<q>` are true
- `xor <p> <q>`, where `<p>` and `<q>` are propositions, which is true if one of `<p>` or `<q>` is false and the other one is true

It is somewhat strange the relatively brainy electorate has not been able to determine the correct character types of the candidates, as that is indeed always possible given the transcript of a debate. The devoted patrons of `dank.nt.nu`, however, believe you'll prove equal to the task.

Input

Input describes one full debate transcript. The first line of input contains two integers N and K, denoting the number of candidates and the total number of utterances in the debate. The candidates are named by the integers 1 to N. The next K lines describe the utterances in the debate, sorted by time; the first such line describes the first utterance of the debate. Each such line consists of an integer, denoting the name of the speaker of the utterance, and the uttered statement, expressed in the language described above. Adjacent tokens on the same line will be separated by exactly one space character.

Output

Output N lines. On the ith line, name the character type (`truther`, `fabulist`, or `charlatan`) of the candidate whose name is i.

Limits and additional notes

- $1 \leq N \leq 7$
- $1 \leq K \leq 100$
- No line of input will be longer than 2049 characters (including the line-feed character).
- All candidates know each other's character types.
- Character type claims will only be about persons in a debate. I.e. for the propositions `truther <n>`, `fabulist <n>`, and `charlatan <n>`, one will always have $1 \leq \texttt{n} \leq N$.

Sample Input 1

```
1 2
1 charlatan 1
1 not charlatan 1
```

Sample Output 1

```
charlatan
```

Sample Input 2

```
2 1
1 and fabulist 1 fabulist 2
```

Sample Output 2

```
fabulist
truther
```

Sample Input 3

```
3 2
1 fabulist 3
3 and truther 1 truther 2
```

Sample Output 3

```
truther
fabulist
fabulist
```

Garbage Tracking

Problem ID: garbagetracking

Talulla has been hired to fix an issue for Sgudal Cleaning - a company that specialises in making garbage cleaning robots for the city New Glasgow. Instead of collecting garbage from the streets, the cleaning robots now *dump* garbage instead! In fact, the cleaning robots now emit a fixed amount of garbage between two intersections. Fortunately, Sgudal Cleaning now has managed to stop all the robots from doing more harm.

To understand the extent of the garbage problem, Talulla has made a program to find out the total amount of garbage there is in a particular area of the city. This looked hard at first, but fortunately, New Glasgow is designed as a perfect grid, and the robots walk in rectangular paths in the city. The sections Talulla is asked to look at are rectangular as well.

Talulla has completed her version of the program, but wants to ensure she has implemented it correctly. She has therefore asked you to implement a version to compare it with.

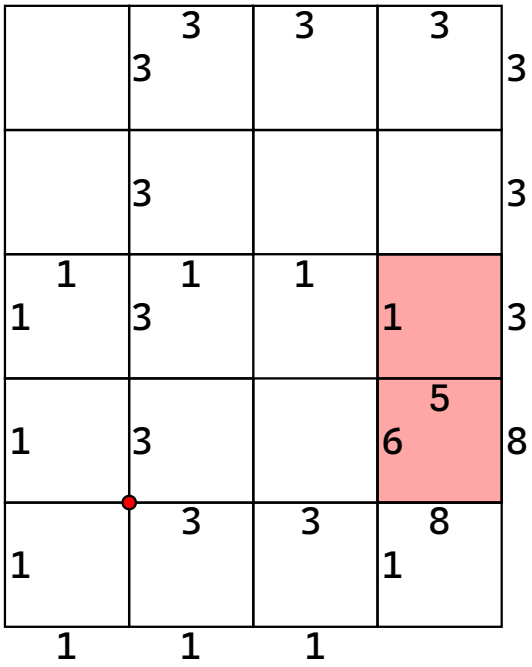


Figure 1: The city grid in Sample Input 1, after the robots have dumped their garbage. The origin is marked by a dot, and the highlighted rectangle represents the last query.

Input

The first line consists of two integers R and Q, the total number of robot movements and queries.

Then follow R lines, each describing a robot's path. Each line contains 5 integers x_i, y_i, w_i, h_i, c_i. A robot has completed a rectangular walk, starting at the intersection at (x_i, y_i). The walk went w_i blocks east, h_i blocks north, w_i blocks west and finally h_i blocks south, dumping c_i kg of garbage between each intersection it visited.

Finally, Q lines follow, one line for each query. Each line contains 4 integers x_i y_i w_i h_i. Sgudal Cleaning wants to know how much garbage there is in the $w_i \times h_i$ block rectangle, where the lower left corner is the intersection at (x_i, y_i).

As the input may be very large, you should consider buffering it.

Output

For each query, print out the amount of garbage in kg there is in the given rectangle, including the edges.

Limits

- $1 \leq R, Q \leq 200\,000$
- x_i, y_i, w_i, h_i and c_i are all integers.
- $-2\,250 \leq x_i, y_i < 2\,250$
- $-2\,250 < x_i + w_i, y_i + h_i \leq 2\,250$
- $0 < c_i \leq 1\,200$

Sample Input 1

```
3 8
0 0 3 4 3
-1 -1 3 3 1
2 0 1 1 5
-1 -1 2 2
0 0 2 2
1 1 1 2
0 0 2 2
1 1 1 3
2 2 1 1
2 0 1 1
2 0 1 2
```

Sample Output 1

```
10
21
2
21
5
3
27
31
```

Helpful Rotations

Problem ID: helpfulrotations

Relatively close to Mars, there is a small group of space habitats shaped as discs. To make movement between them easy, they are all perfectly aligned in the Z coordinate. But while movement is usually easy, there are cases when it's not:

"Hey Sophia, this situation reminds me of that story you told about your father" Galatea said. "Oh, the one where he had issues getting back to our castle? Well.. sure, it was a *kind* of ship, and had steering problems too. I don't care about how many ways we can get to a space habitat with a repair station though" Sophia replied. Galatea had tried to fix their steering engines, but quickly figured that they needed to get to a proper repair station.

Sophia's spaceship is an older model, with some rather weird quirks. For one, the ship must either have its nose or tail faced directly at the habitat's centre to dock. It also has engines on its nose; so while they can't steer in any direction right now, they can accelerate and decelerate at a clicks per second.

"I guess we can jump from habitat to habitat by just waiting, right? The habitats rotate with a fixed speed; at some point we'll point towards some other habitat's centre." she explained. "That'll probably take ages though, and you forget the abandoned habitats: They decided to stop their rotation for safety reasons, right?" Sophia muttered. "I'm not so sure. . . I do recall some habitats spinning really fast, so it may be faster than waiting for the repair bots." Galatea argued. "I highly doubt that, but we'll see. Computer, find the fastest path to a space habitat with a repair station."

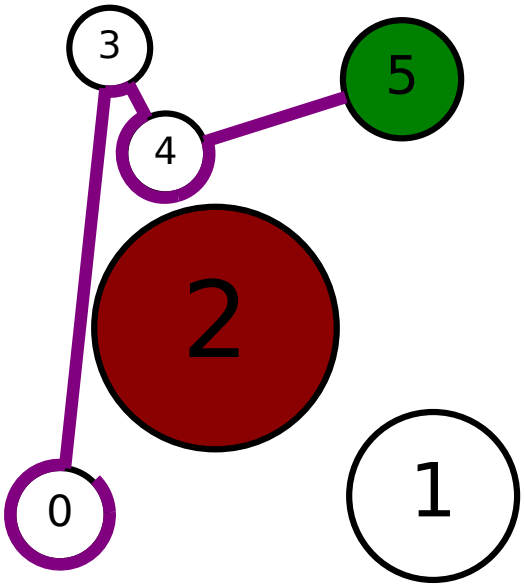


Figure 1: Visualisation of Sample Input 1, and the fastest path to the repair station habitat.

Input

The first line consists of 2 integers followed by 2 real numbers: H, H_{start}, θ_{start} and a. H is the number of space habitats, H_{start} is the habitat Galatea and Sophia are at, and θ_{start} is their current angle in radians. The max acceleration of their spaceship is a clicks per second.

Then follows H lines, one representing each habitat. Each line consists of 4 real numbers x_i, y_i, r_i and ω_i, along with a single character R_i. The real numbers represent a circle with centre at (x_i, y_i) and a radius of r_i clicks, spinning at ω_i radians per second. $R_i = \texttt{t}$ if there is a repair station at the habitat, $\texttt{f}$ otherwise.

Output

Print out the time it takes to get from their current location to a space habitat with a proper repair station. Your answer must have an absolute or relative error of at most 10^{-6}.

If their ship is unable to reach a space habitat with a proper repair station, print out `request repair bot assistance` instead.

Limits and Additional Notes

- $0 \leq H_{start} < H \leq 175$
- $0 \leq \theta_{start} < 2\pi$
- $1.0 \leq a \leq 11.2$
- $0 \leq x_i, y_i < 10\,000$, $1.0 \leq r_i \leq 100.0$
- For all $i \neq j$, $r_i + r_j + 1 \leq \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}$
- $-4\pi \leq \omega_i \leq 4\pi$
- $R_i \in \{\texttt{t}, \texttt{f}\}$
- All real numbers in the input have at most 4 digits after the decimal point.
- Sophia's spaceship is small enough to be considered a point.
- A valid travel path from one space habitat i to j must not intersect with or touch any other habitat.
- Both the docking and undocking procedure is instantaneous. When docking, the spaceship must have zero velocity, and when undocking, the velocity caused by the habitat's rotation is cancelled.
- Floating point inaccuracies for nonzero numbers will not result in wildly different results. More precisely: a relative change in the real number inputs of at most 10^{-8} will result in an absolute or relative change in the accepted answer by at most 10^{-6}.

Sample Input 1

```
6 0 0.78 9.87
0.0 0.0 8.0 -1.3 f
60.0 3.0 14.0 0.3 f
25.0 30.0 20.0 0.0 f
8.0 75.0 7.0 0.2 f
17.0 58.0 7.0 5.0 f
55.0 70.0 10.0 1.0 t
```

Sample Output 1

```
17.62702752357573
```

Sample Input 2

```
4 3 1.0 4.0
0.0 20.0 20.0 1.2345 t
50.0 0.0 30.0 0.0 f
80.0 40.0 5.0 2.1 f
100.0 20.0 20.0 -3.1 f
```

Sample Output 2

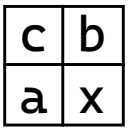
```
12.968636660192796
```

Image Compression

Problem ID: imagecompression

Saskia wants to reduce the bandwidth usage of her website, and has investigated how she can compress her PNG images. She got a bit curious on how PNG image optimizers work, and before she knew it, she started working on her own!

In PNG, an image is considered a grid of Y rows and X columns of bytes. Before the image is compressed, each row is transformed by one of five "filter transform" methods.



In this problem, we'll only consider the first four for convenience. Those filter methods use previously decoded bytes, either from the byte above (`b`) or the value left of this byte (`a`), as shown in the image above.

The four filter methods we are concerned with are defined as follows:

Name	Function
None	`Filt(x) = Orig(x)`
Sub	`Filt(x) = Orig(x) - Orig(a)`
Up	`Filt(x) = Orig(x) - Orig(b)`
Average	`Filt(x) = Orig(x) - floor((Orig(a)+Orig(b))/2)`

As we're dealing with bytes, be sure to do unsigned modulo 256 immediately after addition and subtraction to get the right result. In addition, if `a` or `b` doesn't exist, they are considered to be 0.

A common heuristic when choosing which filter to use is to pick the methods that give as many zero bytes as possible after the transformation. However, Saskia believes it is much better to use the filter methods that gives the most of ANY particular byte instead. Can you help her make a program that finds the optimal filter transforms?

Input

On the first line, you are given two positive integers, Y and X, denoting the number of rows and columns of the image, respectively.

Then follow Y lines, each with X space-separated integers $b_{y,x}$, representing the picture.

Output

Print the byte that will have the highest frequency if the optimal filter methods are used, along with its frequency. If there are multiple bytes with the highest frequency, print the highest one.

Limits

- $0 < Y, X \leq 350$
- $0 \leq b_{y,x} < 256$

Sample Explanation

In Sample 1, we can use the filters Up, Sub, Up and Average to get 4 occurrences of 73. We can also use Sub, Up, None and Average to get 4 occurrences of 87. As 87 is the biggest, we print that one.

Sample Input 1	Sample Output 1
4 7 251 111 206 182 13 242 73 215 93 166 13 185 159 211 43 166 218 255 48 157 87 94 92 118 142 196 206 1	87 4

Jumbo Javelin
Problem ID: jumbojavelin

Photo by Álvaro Pérez Vilariño, cropped and flipped

Jessica wants to become a javelin thrower, moreover she wants to become a famous one. However, she doesn't want to train 40 hours a day. To avoid much training while still becoming famous, she has concocted a plan: make all her javelins gigantic.

As a software developer by day, she knows she needs a minimum viable javelin for testing, but also that you can glue existing things hastily together and get something that works alright.

She has a couple of steel rods next to her bed – as one does – and decides to go to the blacksmith apprentice Jack to get him to fuse them together.

The good thing about Jack is that he'll fuse the steel rods together for free. But Jack isn't that good at fusing things together. When fusing two rods, the fused rod will lose 1 cm of their combined length.

That's not a problem for Jessica, but she wonders how big her javelin will become if Jack fuses all the rods together.

Input

The first line consist of an integer N, the number of steel rods she has. Then follows N lines, each containing a single integer l_i representing the length of the steel rod in cm.

Output

Print out the length of the jumbo javelin Jack has made for her.

Limits

- $1 < N \leq 100$
- $1 \leq l_i \leq 50$

Sample Input 1

```
4
21
34
18
9
```

Sample Output 1

```
79
```

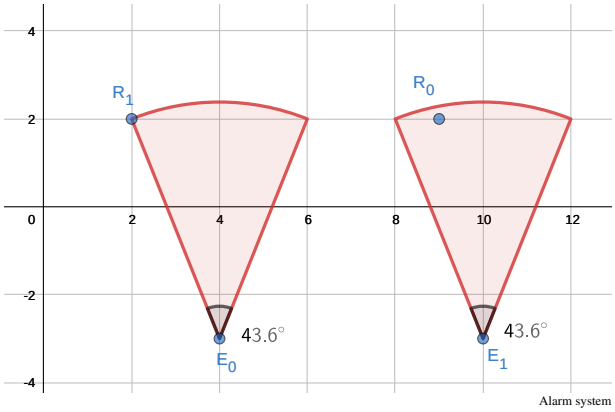
Sample Input 2

```
3
1
50
40
```

Sample Output 2

```
89
```

Knox
Problem ID: knox



Alarm system

Ellen has just bought herself a new fort to protect all her precious belongings. Her fort is of course very well protected, and in fact the only possible point of entry is the main hallway. Ellen's hallway is a big room with a long and narrow path through the center (aligned with the x-axis). The narrow path is the only place possible to walk, as both sides of the path are guarded by an electric fence. However, Ellen feels that this is not enough, and wants to invest in a motion detection alarm system to protect the hallway.

The alarm system she wants to buy consists of laser emitters and reflectors that can be paired together. The emitters send out a laser in a sector with a width of a degrees and a unique frequency f_i. If a reflector is placed inside this sector, set to the correct frequency and turned exactly towards the emitter, the laser will be reflected back to the emitter, and a barrier is formed. The emitters, reflectors and brackets are all made of a special transparent material such that the lasers pass straight through them, with the only exception being laser and the reflector tuned to the same frequency.

Fortunately, the previous owner had already bought such an alarm system, but he took the emitters with him when he left. Hence, the reflectors and brackets for the emitters are still standing. All the brackets for the emitters are placed on one side of the room, such that the center line of the sector is normal to the x-axis. Ellen's task is to buy new emitters, turn the reflectors to face each of the emitters and set them to the corresponding frequency.

In the hasty bidding process for her new fort, Ellen spent a bit more money than she would have liked. Now she would want to reuse the alarm setup that already exists, and spend as little money as possible on her new laser emitters. The price of the emitters is directly proportional to the width of the laser sector, and the manufacturer also adds a huge fee for ordering emitters with different widths. So given that all the emitters have the same sector width, what is the smallest sector width a she can order for the emitters and still be able to connect all the emitters to a reflector?

Input

On the first line, you are given one positive integer, N denoting the number of emitters and reflectors.

Then follow N lines, each with two space-separated numbers, x_i and y_i, representing the coordinate of the i'th laser emitter bracket.

Then follow N more lines, each with two space-separated numbers, x_j and y_j, representing the coordinate of the j'th reflector.

Output

One number, a, the minimal sector width (in degrees) necessary to connect each reflector to exactly one emitter. Your answer must have an absolute or relative error of at most 10^{-6}.

Limits

- $0 < N \leq 300$
- $-1\,000 \leq x_i, x_j \leq 1\,000$
- $-1\,000 \leq y_i < 0$
- $0 < y_j \leq 1\,000$
- All real numbers in the input have at most 4 digits after the decimal point.

Sample Explanation

Sample Input 1 is shown in the figure at the top of the problem.

Sample Input 1	Sample Output 1
2 4.0 -3.0 10.0 -3.0 9.0 2.0 2.0 2.0	43.6028189727036

Lemonade
Problem ID: lemonade

Simon just wanted to make cute robot dogs to keep as pets, but they turned against him. They have currently taken over his garden — where they are preparing for world domination. The robot dogs are (obviously) fueled by lemonade, and they have built a small lemonade factory using lemons from Simon's garden. Simon needs to act now in order to stop them from producing enough lemonade for world domination!

Unfortunately, the robot dogs guard the lemon trees at all times and are patrolling the factory at regular intervals. This makes it hard for Simon to sabotage, but after having observed them for a while he might have found a way. Every ten minutes there's a small time window in which a tap to the main mixing tank is out of sight of the patrolling dogs. In that small window of time Simon can run up to the mixing tank with a hose and empty the tank. Once the tap is opened it can't be closed before the tank is emptied. The mixing tank can be assumed to not be filled or emptied by anyone else during this small window of time. Even when faced with the threat of robot dog world dominance, Simon doesn't want any good lemons to go to waste. Hence, he will empty the content of the mixing tank into his own tank inside his house, where it will be safe from the robot dogs. This tank will initially be empty and have a capacity of N liters.

From his previous scouting, Simon have figured out that he has M consecutive opportunities to empty the mixing tank. The dogs have unlimited amount of water, so the goal is to steal as much lemonade juice as possible. The content of the mixing tank changes regularly as part of a complex lemonade production process. Let v_i and l_i be the total volume of the content and the amount of lemon juice respectively in the mixing tank for the ith ($1 \leq i \leq M$) opportunity to empty it. The volume v_i is in liters and l_i is in number of lemons worth of juice. Then the volume and amount of lemon juice changes according to

$$v_i = ((a * i + b * v_{i-1}^*) \bmod c) + 1$$
$$l_i = (d * i + e * v_{i-1}^*) \bmod f$$

Where a, b, c, d, e, f are constants, while v_i^* is the content volume Simon has or has not emptied the mixing tank at the ith opportunity. If the mixing tank was not emptied we simply get $v_i^* = v_i$, but if it is emptied we get $v_i^* = 0$. In other words, whether Simon chooses to empty the mixing tank or not at any given opportunity will affect the content volume and amount of lemon juice at the next opportunity.

Simon wants you to help him find out at which opportunities he should empty the mixing tank to maximize the amount of lemon juice he will steal from the robot dogs, while at the same time respect the capacity of his in-house tank.

Input

The first line will contain two positive integers, N and M. And the second line will contain eight positive integers, a, b, c, d, e, f and v_0^*.

Output

Print the maximal amount of lemon juice Simon is able to steal from the robot dogs.

Limits

- $1 \leq N, M, c, f \leq 150$
- $1 \leq a, b, d, e \leq 1\,000\,000$
- $0 \leq v_0^* \leq c$

Sample Explanation

In Sample 1, at Simon's first opportunity, there is $(7{\times}1{+}1{\times}0) \bmod 10{+}1 = 8$ liters of content and $(1{\times}1{+}1{\times}0) \bmod 11 = 1$ lemons worth of juice. If he chooses to empty the tank, there will be $(7 \times 2 + 1 \times 0) \bmod 10 + 1 = 5$ liters of content at next opportunity. And so he wouldn't be able to empty the mixing tank the second time because $7 + 5 = 12$ goes beyond the capacity of his in-house tank. Ending up with only 1 lemon worth of juice. On the other hand, if he

does not empty the mixing tank at the first opportunity, the mixing tank will have $(7 \times 2 + 1 \times 8) \bmod 10 + 1 = 3$ liters of content and $(1 \times 2 + 1 \times 8) \bmod 11 = 10$ lemons worth of juice at the second opportunity. So the optimal plan is to wait out the first opportunity and then empty the one liter containing 10 lemons worth of juice at the second opportunity.

Sample Input 1	**Sample Output 1**
10 2 7 1 10 1 1 11 0	10