FAU Winter Contest 2016

January 30th

acm International Collegiate Programming Contest

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Problems

A Alternative Typesetting
B Apfelschorle
C Boring Conversation Anyway. . .
D Eating Stones
E Jedi Recruiting
F Little Elephant *(easy)*
G Meet Again
H R2-D2's Lovely Voice
I Random Battle Droid
J Rubik *(very easy)*
K Stormtrooper Dating
L We Like Trains *(easy)*

Sponsored by

accenture
High performance. Delivered.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

This page is intentionally left (almost) blank.

Problem A: Alternative Typesetting

A long time ago, one did not use a fictitious story on the problem statement but directly provided the . . .

Input \

But nowadays, we are in a period in which everybody wants the problem to be introduced by nicely aligned text. That is, the word wraps are automatically inserted making the lines as similar as possible. However, one issue remains . . .

The issue is, *what* is meant by similar. The reading eye is only pleased if the text appears to be homogenous. This can be formalized by the *badness of a line*. The badness of a line of text containing m words w_i is calculated as:

$$\text{badness} = b_\ell + \sum_{i=0}^{m-2} |(\text{number of spaces between } w_i \text{ and } w_{i+1}) - 2|^2$$

where b_ℓ is a constant, and the overall badness is defined as the sum of badness over all lines of text.

No justified line is allowed to have leading or trailing spaces and words must always be separated by at least one space.

Artistic freedom allows for replacing words with synonyms.

Input

The first line contains two integers, the column width w and the constant b_ℓ. The next line contains the number of synonymous word sets Y. Then follow Y disjunct sets of synonymous words, each on one line, described by its size s and the words it contains.

The next line contains N, and N words in the line after that.

N is guaranteed to be even, $0 < N < 1000$, $0 \leq Y < 1000$, $0 < s < 10$, $0 < w \leq 80$, $0 \leq b_\ell < 10^6$. All words in the input are guaranteed to consist only of ASCII letters, digits and punctuation marks (’.’, ’,’, ’!’, ’?’). All words fit at least twice in a single line (with one space between them).

Output

Output the input words justified so that the overall badness is minimized. If there are multiple possibilities, output any of them.

Hint: The minimum possible badness for sample 1 is 32.

Sample Input 1

```
22 10
2
3 best greatest worldclass
2 greatest! worst.
12
The best of the best of the best will be the greatest!
```

Sample Output 1

```
The  greatest  of  the
greatest of  the  best
will  be  the   worst.
```

Problem B: Apfelschorle

Competetive programmer Mike is flying to a prestigious programming contest. He's thirsty and wants an Apfelschorle (also known as "apple juice spritzer"; it consists of apple juice mixed with carbonated mineral water).

He is not sure whether the flight attendant knows the word "Apfelschorle". So he orders an "apple juice with water" instead.

The flight attendant hands him two plastic cups, one with pure apple juice, one with carbonated water. Before Mike can clarify, the flight attendant is already serving the next row.

Mike really loves Apfelschorle, so he needs to fix this. To avoid spilling, flight attendants never fill a cup completely, so Mike is able to pour some liquid from one cup into the other, back again and so on.

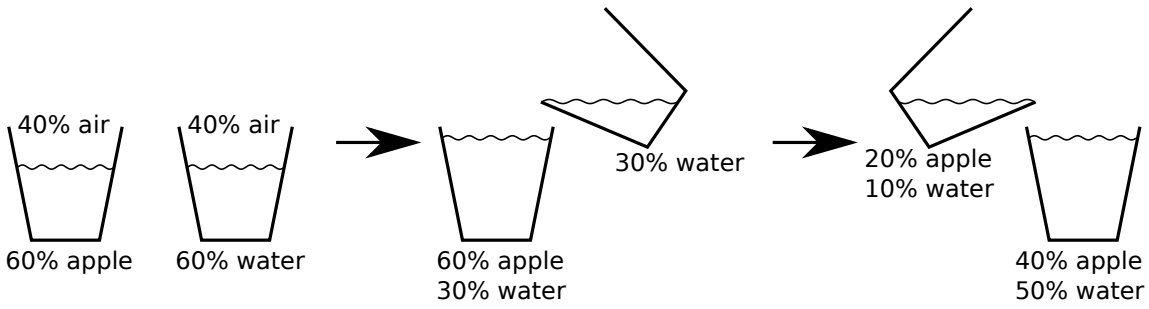


Figure B.1: Example: Mike gets two cups, pours twice.

Continuing infinitely, he would end up with the same ratio of components in both cups. But Mike does not want to spend an infinite amount of time. However, he still wants to achieve reasonably similar ratios in both cups.

Mike defines the ratio R_X of cup X as $\frac{\text{amount of apple juice in cup } X}{\text{total amount of liquid in cup } X}$. He thinks reasonably similar ratios are achieved when $|R_1 - R_2| \leq \frac{D}{10^6}$, where D is the precision Mike wants to achieve and depends on his mood.

Can you help him to determine the minimum number of times he has to pour liquid from one cup into the other to reach "reasonably similar ratios"?

You may assume the following: Both cups are of same size. Liquids mix instantly. Mike does not spill liquid on accident or purpose. If a cup X is empty after pouring, R_X equals R_Y. Mike does not drink from any cup before "reasonably similar ratios" are achieved.

Input

There is only one line of input. It contains three integers A, W and D ($1 \leq A, W \leq 99$; $10^3 \leq D < 10^6$). The first cup is filled to A percent with apple juice. The second cup is filled to W percent with carbonated water. D refers to the constant in Mikes formula for "reasonably similar ratios".

It is guaranteed that adding or subtracting 1 to D does not change the result.

Output

Print a single line containing one integer, the minimum number of times Mike has to pour liquid from one cup into another, so that the formula for "reasonably similar ratios" is true.

Sample Input 1

```
60 60 300000
```

Sample Output 1

```
2
```

Sample Input 2

```
50 80 200000
```

Sample Output 2

```
2
```

Problem C: Boring Conversation Anyway. . .

Heavily armed, our friends Han, Luke, and Chewbacca have just broken into the prison of the Imperial Space Station in order to rescue the allied Princess Leia. While Luke is searching for the princess' cell, Han holds the line at the prison's entrance. Unfortunately, an imperial officer far away became suspicious after noticing the noise of the shots in the prison and directly made a phone call to the entrance, where Han is waiting. Taking the call, Han is in trouble: what to say in order to provide Luke with as much time as possible?

Figure C.1: Han on the phone

Luckily, the officer's questions to Han are predictable, because they strictly follow the Imperial Communication Protocol Convention (ICPC). This protocol regulates the questions to ask and how to react to answers: the convention specifies a set of possible answers and for each a subsequent question.
Unfortunately, for some questions the convention does not offer any predefined choice, e.g. for personal questions like in Fig. C.2. In these cases, Han has no other choice than aborting the conversation, resulting in the officer raising the alarm.

Officer: What happened?
Han: Slight weapons malfunction, but we're fine.
Officer: We're sending a squad up.
Han: Negative. We had a reactor leak here now. Give us a few minutes . . .
Officer: Who is this? What's your operating number?
Han:

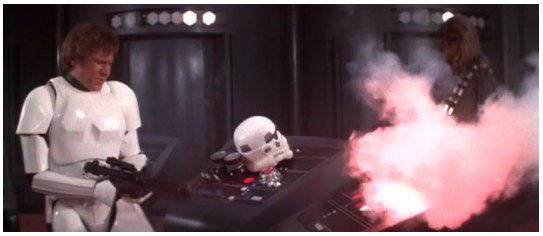


Figure C.2: An example conversation ending with a personal question.

If the officer is already satisfied and not suspicious any more, he starts asking so-called safe questions. Hence, after those questions Han can safely end the conversation and hang up. But he can also continue the conversation if he likes to. Given the ICPC, is it possible for Han to keep the conversation going on forever; and if not, how many safely ending conversations are there?

Input

The input starts with a line containing two integers Q and S ($1 \leq Q \leq 10\,000$, $0 \leq S \leq Q$). Q denotes the number of questions and S the number of safe questions. The second line contains S distinct integers h, with $0 \leq h < Q$, listing the questions on which Han can safely end the conversation if he likes to.

Then Q lines follow. Each of them begins with an integer $0 \leq n \leq 10\,000$, denoting the number of answers that are offered by the ICPC, followed by n integers $q_1, \ldots, q_n$, with $0 \leq q_i < Q$. If Han picks the ith answer, then the officer will ask the question q_i next. The officer will start the conversation by asking question 0.

You may safely assume that the sum of all the ns will not exceed $1\,000\,000$.

Output

Print `infinite`, if Han is able to keep the conversation going forever. Otherwise, print the number of safely ending conversations modulo $1\,000\,000\,007 = (10^9 + 7)$. Two conversations are considered the same if they consist of the same questions and answers in the same order.

Sample Input 1

```
5 2
3 4
2 1 2
2 4 4
2 3 4
0
0
```

Sample Output 1

```
4
```

Sample Input 2

```
5 2
2 4
2 1 3
2 2 3
1 1
1 4
0
```

Sample Output 2

```
infinite
```

Sample Input 3

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

Sample Output 3

```
infinite
```

Problem D: Eating Stones

Luke is training his mind with Yoda, they are playing a game called “Family-Drama”.

Yoda sets up the board of 3×4 fields. He has two kinds of stones: black ones, which apparently stand for dark-sided camps, and white ones, for rebellion camps. He places a few of those stones on the board, not more than one per field, and leaves the rest next to the board.

“Battlefield, this is and go to war for the rebellion, you must.”

The result of this game is the number of rebellion camps minus the number of dark-sided camps on the board after the game ends. Luke’s goal is to maximize the result and Yoda’s goal is to minimize the result.

Yoda continues to explain the rules:

- They both take alternating turns, Luke placing the leftover white stones, Yoda the black. In every turn, a player has to place a new stone on an empty field, he may only pass if he cannot make a valid move or if he has no more stones left.
- The game ends if none of the players can make a **valid** move any more. A move is considered **valid** if one of the leftover stones is placed on an empty field and the move is not **suicidal**.
- Let us call camps connected horizontally or vertically a *group* of camps. A group is alive if at least one of its camps is connected horizontally or vertically to an empty field (border does not count as empty). Otherwise it is dead.
- Dead groups are removed from the board after each turn and Luke eats the stones, they may not be placed again.
- A move is **suicidal** if it would result in one of the player’s own groups being dead (before stone removal).

After losing a lot and eating even more stones Luke gets angry, and wants to know the purpose of this game.

“In war you fight only the battles, win, you can. Hmmm.”

Can you help Luke find out how the game will end once Yoda set up the board assuming they both play optimally?

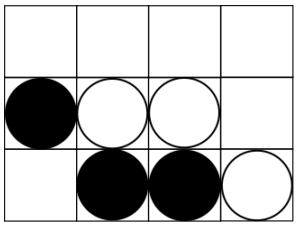


Figure D.1: Placing a white stone in the bottom left corner is suicidal.

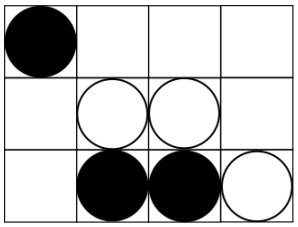


Figure D.2: Luke may place a stone in the bottom left corner to kill the two black stones.

Input

One line with two integers: The number of white and black stones, they will be between 0 and 8, inclusively. Three lines with four characters each: '`.`' for an empty field, '`x`' for dark-sided camps and '`o`' for rebellion camps. It is guaranteed that every group of camps in the initial field is alive.

Output

Print the result of the game if both play optimally and Luke moves first. They play until none of them can make a valid move any more and then count the stones on the board to compute the result.

Sample Input 1

```
8 8
xxxo
xxoo
x.oo
```

Sample Output 1

```
-1
```

Sample Input 2

```
5 0
x.x.
...x
x.x.
```

Sample Output 2

```
2
```

Sample Input 3

```
4 1
....
..x.
.x..
```

Sample Output 3

```
1
```

Problem E: Jedi Recruiting

The Jedi are always searching for gifted children, who feel the force. But before the children are taught they face various challenges, to test their abilities. One of those tests requires a marble board.

The marble board is a simple wooden plank, which is positioned in an inclined manner. There are various grooves in the plank where marbles can be placed. The grooves form many intersections and therefore the marbles can take different routes. The task for the Jedi applicant is then to predict the route of the marbles.
To increase the difficulty for that test the supervisors decide to create new marble boards.
They specify quite strict instructions to create specific marble boards. On top there should be a single groove - the start of the marbles. There will be splitting and joining intersections.
A splitting intersection has one input groove and two output grooves. The marbles may run into this intersection only through the input groove. The marbles will then leave the intersection either through the left output groove or through the right output groove.
A joining intersection has two input grooves and one output groove. The marbles may run into this intersection either through the left input groove or through the right input groove. The marbles will then leave the intersection through the single output groove.
All grooves will either end as an input for an intersection or they end on the bottom of the board. Intersections are specified in order from top to bottom. No two intersections will be at the same

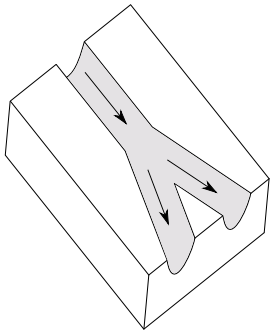


Figure E.1: Splitting intersection

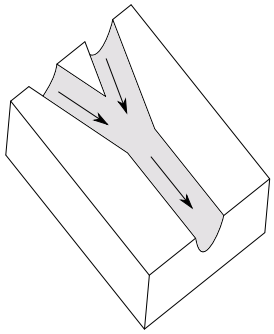


Figure E.2: Joining intersection

height. Marbles only run downwards.
Please help the supervisors and verify the validity of their specification by checking whether the grooves can be placed without crossings.

Input

The first input line contains the two integers G the number of grooves and I the number of intersections ($1 \leq G, I \leq 200$). The index of the starting groove is 0. Each of the next I lines specify an intersection in order from top to bottom. The specification of an intersection consists of a character followed by three integers. If the character is an 'S' then a splitting intersection is specified and the three integers are i the index of the input groove, o_1 the index of the left output groove and o_2 the index of the right output groove ($0 \leq i, o_1, o_2 < G$). If the character is a 'J' then a joining intersection is specified and the three integers are i_1 the index of the left input groove, i_2 the index of the right input groove and o the index of the output groove ($0 \leq i_1, i_2, o < G$). Each index of a groove will appear at most once as input and once as output and it will appear as output not before it will appear as input. The only exception is the starting groove with index 0, which will appear only as the input of the first intersection.

Output

Print `valid` if there is a valid placement for the given specification and print `invalid` otherwise.

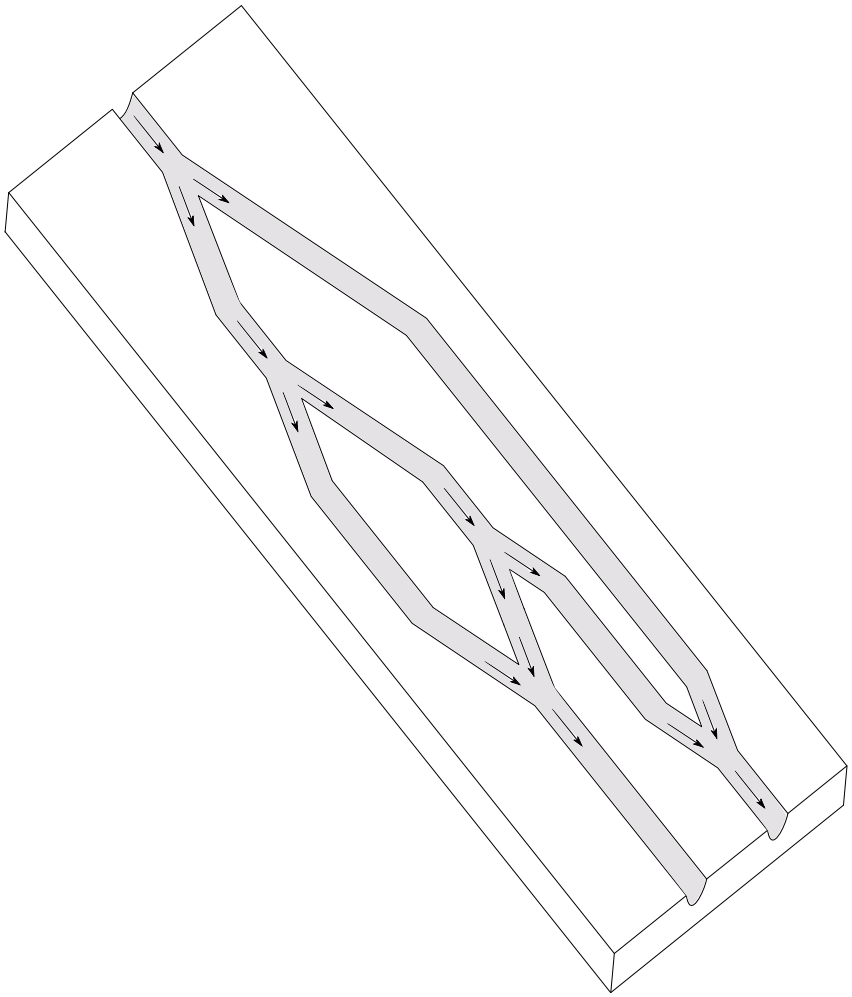


Figure E.3: Visualization of a valid placement for the given specification in the first sample input

Sample Input 1

```
9 5
S 0 8 2
S 8 3 4
S 4 5 6
J 3 5 7
J 6 2 1
```

Sample Output 1

```
valid
```

Sample Input 2

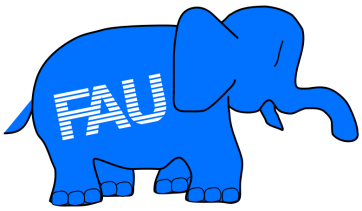
```
4 2
S 0 1 2
J 2 1 3
```

Sample Output 2

```
invalid
```

Problem F: Little Elephant

After the galactic empire has been defeated, a lot of rebels who previously spent their whole time fighting the empire had to look for a new profession. Gardeners and zookeepers were particularly interesting and relaxing professions for many of them.



Little eleFAUnt has garden. You help!
Image is public domain, based on
http://openclipart.org

Former rebel Jan is doing both: He bought a farm on the planet Anobis, and is taking care of a lot of animals. For his special favourite, the little elephant Karl, Jan built an entirely new garden, consisting of some trees (that bear some tasty fruits for his elephant) and a few fences. After Jan let the little elephant enter the garden, he noticed that not all of the trees can be reached by the little elephant.

You are given a map of the garden, consisting of the following symbols:

- `#` symbols a fence. As the little elephant is still quite small, he cannot cross fences
- `.` is grass. The little elephant can walk freely over it
- `E` is the current position of the elephant
- `*` is one of the trees the elephant is trying to reach. The trees are high, and the elephant can easily walk beneath them.

Help Jan to find out how many of the trees can be reached starting from the current position of the little elephant. While walking, the little elephant can do steps to the left, right, up and down, but not diagonally, thus he cannot cross diagonal fences (see the third sample input).

Input

The first line contains two integers H and W ($1 \leq H, W \leq 100$), the height and the width of the following map of the garden. H lines follow, each containing W characters describing the garden with the symbols explained above. The character `E` occurs exactly once per input file.

Output

A single integer, the number of trees that the little elephant can reach from his current position.

Sample Input 1

```
5 5
#####
#.*.#
#.E*#
#*..#
#####
```

Sample Output 1

```
4
```

Sample Input 2

```
4 5
*....
.....
.E...
....*
```

Sample Output 2

```
2
```

Sample Input 3

```
3 3
#**
.#*
E.#
```

Sample Output 3

```
0
```

Problem G: Meet Again

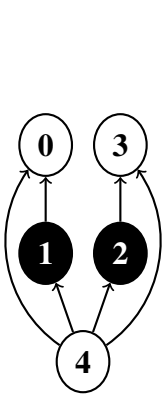
While hiding from the evil empire, the spaceship *Millenium Falcon* lands at the entrance of a cave network within an asteroid. However, the cave walls do not seem to be made out of stone, but feel warm and soft. The crew tries to investigate the cave they have landed in. So they go deeper and deeper into the cave network and eventually split up. Only the pilot Chewbacca stays at the spaceship.

After some time, one of the crew members realizes in what kind of cave network they are: in the stomach of a giant monster! In order not to be digested, he tells Chewbacca to use the spaceship to pick up the crew members that are still distributed over the cave network.

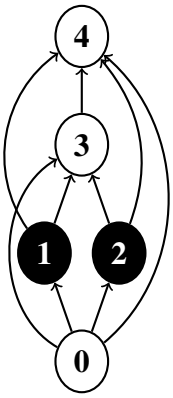
Because landing and starting the spaceship wastes a lot of time, they plan to meet at a single point in the cave network, where all crew members enter the ship again:

1. A meeting point must be safely reachable by *all* crew members. That means, any crew member is only allowed to stay where he is or walk through tunnels going strictly upwards the cave network.
2. A meeting point must not be unnecessarily far upstairs. That means, there must be no other cave more downstairs that is still safely reachable by all crew members.

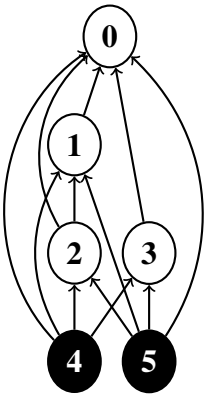
Knowing which cave is safely reachable from which one and the crew members' position, can you help them whether there is a unique cave to meet and which one it is?



Sample Input 1



Sample Input 2



Sample Input 3

Input

The first line of input starts with an integer $1 \leq V \leq 1\,000\,000$, the number of caves and an integer $0 \leq E \leq 1\,000\,000$ denoting the number of upward-going tunnel paths and one integer $1 < S \leq V$, the number of crew members. The second line contains S pairwise distinct integers $0 \leq s_i < V$, denoting the cave of the ith crew member.

Then E lines follow, each of them consisting of two integers $0 \leq a, b < V$, $a \neq b$, telling that it is possible to safely reach b from a, or equivalently that there is an upward-going tunnel path from a to b. If b is safely reachable from a, and c is safely reachable from b, then c is safely reachable from a, i.e. (a, c) is also listed in the input.

Output

Tell, whether there is a unique meeting point. If there is no cave that is safely reachable by all, print `none`. If there is a unique meeting point, then print its index $0 \leq i < V$. If there are

multiple meeting points, print `multiple`.

Sample Input 1

```
5 6 2
1 2
1 0
4 0
2 3
4 3
4 1
4 2
```

Sample Output 1

```
none
```

Sample Input 2

```
5 9 2
1 2
0 1
0 2
1 3
2 3
3 4
2 4
0 3
0 4
1 4
```

Sample Output 2

```
3
```

Sample Input 3

```
6 12 2
4 5
1 0
2 1
2 0
3 0
4 2
4 3
5 2
5 3
4 0
5 0
4 1
5 1
```

Sample Output 3

```
multiple
```

Problem H: R2-D2's Lovely Voice

R2-D2 is invited to Han Solo's and princess Leia's wedding. Leia is very much in love with R2-D2's voice, so she asked him to sing the famous *song which does never repeat*, a famous mining-robot worker song, while they walk to the altar.

For the song to be as exciting as possible, it is constructed as follows:

- The singer writes down all the different sounds he can make and sings them in an arbitrary order, that is the first verse.
- Every following verse is constructed by its predecessor, a songrule determines the new position of each sound.

Consider the examplary songrule $(3, 1, 4, 2)$: the first sound of a verse is the third sound of the next verse, the second sound is the first in the next verse and so on.

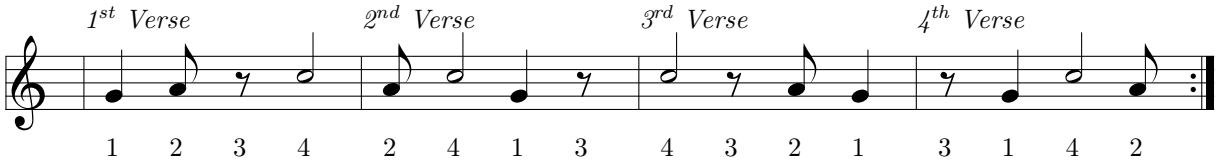


Figure H.1: The song generated by the songrule $(3, 1, 4, 2)$

Sophisticated C-3PO is not as enthusiastic about this song as everyone else, he believes it will always repeat at some point. R2-D2 has been practicing for hours. He sings one verse, makes a break of one second and then starts with the next verse. C-3PO has been listening long enough to figure out the rule he is using, he wants to tell R2-D2 how many verses it takes until the song repeats itself. Maybe that will convince him that this song is a product of overrated pop-culture and he will stop. Can you help?

Input

The first line contains one integer n ($2 \leq n \leq 1\,000\,000$), the different sounds of R2-D2. The second line contains n distinct integers p_i ($1 \leq p_i \leq n$), the songrule: the i-th sound of each verse will be the p_i-th sound of the following verse.

Output

One line with one integer, the number of verses until the song will repeat itself. This number is guaranteed to be at most 10^{18}.

Sample Input 1

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

Sample Output 1

```
6
```

Sample Input 2

```
4
3 1 4 2
```

Sample Output 2

```
4
```

Sample Input 3

```
10
5 3 7 6 4 1 9 10 8 2
```

Sample Output 3

```
12
```

Sample Input 4

```
2
1 2
```

Sample Output 4

```
1
```

Problem I: Random Battle Droid

Battle Droids look all same to you? That made the IFF (Identification, friend or foe) devices of the opposing party, the Galactic Republic, very easy to build. The lead Battle Droid constructor got aware of this and decided to randomly change the look of the Battle Droids. He invented 5 different kinds of torsos, 3 different kinds of arms, and so on. As any robots, the battle droids are produced in one single long conveyor with different production stages. In each stage one decision out of a number of choices can be made. To make this decision random the factory uses dice as they do not trust random number generators. So in the end the look of the battle can vary very much.

Figure I.1: The Battle Droid army

Figure I.2: Different flavors of Battle Droids (Images by jasonedmiston on deviantart.com)

The republic got totally caught pants down in the first fight against the first random Battle Droid army. So jedi master Yoda decided to manipulate the production line of the battle droids to reduce the number of possible different combinations of battle droids. He sends a number of jedi into the production. The jedi can manipulate the dice of consecutive production stages to roll always the same number.

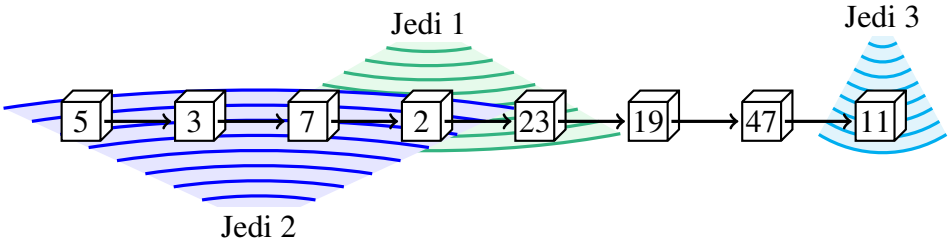


Figure I.3: Yoda's third plan in Sample Input 1

Yoda has different plans how the jedi can position themselves to be able to control different consecutive stages. But calculate the resulting number of different combinations of Battle Droids you must.

Input

The first line contains two integers C and S, where C denotes the number of production steps, and S the number of plans ($1 \leq C \leq 10^6$; $1 \leq S \leq 10^5$). The next line contains C integers c_i, where c_i denotes the number of options for the ith production step ($1 \leq c_i \leq 10^8$).
Then follow S plan descriptions, each in two lines. The first line of the plan description contains an integer W, the number of jedi involved in that plan ($1 \leq W \leq 5$). The second line contains W ranges of fixed choices, each specified with two integers f_j and t_j, where f_j denotes the start of the range and t_j the end of the range (both inclusive, $0 \leq f_j \leq t_j < C$). Note that these ranges may overlap.

Output

Print one line for each plan containing the number of different Battle Droids modulo $1\,645\,333\,507$.

Sample Input 1

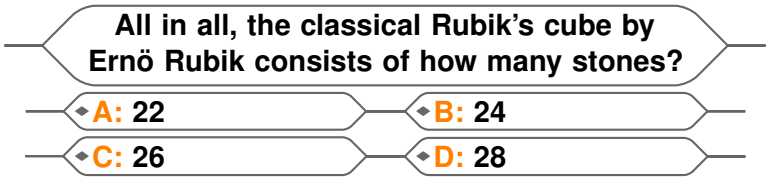
```
8 3
5 3 7 2 23 19 47 11
1
3 5
2
0 0 4 7
3
2 4 0 3 7 7
```

Sample Output 1

```
54285
42
893
```

Problem J: Rubik

Your dear friend Leon is participating in a quiz show where he could potentially win a lot of money, if only he could answer the final question:



Leon knows quite a few things that could help solve this problem. He knows that the classical Rubik's cube has the shape of a cube of edge length n, where the surface is assembled from stones of unit cube size, wired together on the inside with something that is not considered to be stones.

Using this knowledge the question might seem easy to you, but unfortunately Leon is not that good at maths, so he calls you. Can you help him?

Input

Input consists of five integers $n, a_{\mathrm{A}}, a_{\mathrm{B}}, a_{\mathrm{C}}, a_{\mathrm{D}}$ on one line, where n describes the size of the Rubik's cube and a_{X} are the four different possible answers. All given answers will be positive integers $\leq 10^9$ and $0 < n \leq 19$. It is guaranteed that there is exactly one right answer.

Output

Output the letter corresponding to the right answer (`A`, `B`, `C`, or `D`).

Sample Input 1

```
3 22 24 26 28
```

Sample Output 1

```
C
```

Sample Input 2

```
1 2 1 3 4
```

Sample Output 2

```
B
```

This page is intentionally left (almost) blank.

Problem K: Stormtrooper Dating

Stormtrooper cadets at the imperial acadamy train hard to stand out of the crowd. Being allowed to wear the fashionable uniform of a Snowtrooper, Sandtrooper, Scout or even Phase II dark trooper is their ultimate goal.

Darth Norbert after he switched to the dark side.

The hardest exercises for them are the infamous aiming courses set up by Darth Norbert. As everyone knows, Stormtroopers do not aim very well. To pass the aiming exercises, you have to hit a certain percentage of all the targets set up in the different aiming sessions throughout the year. Usually, these sessions are done in groups of two cadets, but one might do them alone as well for the challenge.

Darth Norbert soon found out that this gives him a great business model: similar to a dating platform, he offers a service to find a matching team partner for you. If you pass the course and wouldn't have passed it when doing it alone, Darth Norbert gets money from you. If you help a cadet to pass and would have been able to pass on your own, the platform gives you some money.

Let us assume that many users registered at the platform already, some searching for help, the better cadets looking for money. Cadets need to have an average individual score of at least N percent over all targets. If targets were hit by a team of two the achieved score counts for both individual scores. At the current point of time, D percent of the aiming sessions are already over, and each cadet has hit a certain percentage of the targets so far.
It is assumed that a cadet who hit p percent of the targets up to this point will also hit p percent of the remaining targets. If, instead, two cadets form a team and their hit percentages are p_a and p_b percent respectively, their hit percentage for the remainder of the course will be $\min\left(\frac{p_a+p_b}{2}+T, 100\right)$ percent. T is a "team improvement coefficient" determined by Darth Norbert.

Can you match the users in a way that the "dating" platform makes the most money? It is not necessary that all cadets get assigned a partner and some cadets may fail the course in the end. However, Darth Norbert's platform guarantees its users that they will not fail the course if it manages to find a partner for them, so everybody who gets a partner also has to pass the course.

Input

The first line contains three integers U, N, and T, where U denotes the number of users of the platform, N denotes the percentage of targets to hit and T denotes the percentage a team improves because of working as a team ($1 \leq U \leq 10^6$; $1 \leq N \leq 100$; $0 \leq T \leq 50$). The second line contains three integers G, P, and D, where G denotes the money the platform earns for cadets that pass the course because of the platform. P denotes the money the platform pays if you help a cadet to pass the course and D is the percentage of targets already done in the course ($1 < G \leq 10\,000$; $1 \leq P < G$; $1 \leq D \leq 99$). The last line contains U integers p_i, where p_i denotes the percentage of hit targets so far for student i ($0 \leq p_i \leq 100$).

Output

One line containing the amount of money Darth Norbert can make using the optimal matching of cadets to teams.

Explanation of the Sample Input

In the first case, only a team consisting of the second and third cadet will be able to pass. Pairing them up earns the platform $2 * 21 = 42$. In the second case it optimal to match up the first cadet with the fourth, and the second with the third. The net profit of the platform is $3 * 21 - 11 = 52$.

Sample Input 1

```
4 60 5
21 11 70
58 59 59 58
```

Sample Output 1

```
42
```

Sample Input 2

```
4 60 5
21 11 70
58 59 59 62
```

Sample Output 2

```
52
```

Problem L: We Like Trains

After last year's iteration of the NWERC regional contest, the three FAU teams ("little eleFAUnt has array", "ACK" and "I LIKE TRAINS") went back from Linköping to Nürnberg by plane via Amsterdam. Due to a strong storm, the landing approach to Amsterdam took a lot longer than expected. While some passengers noted that going by train would obviously have been a lot more comfortable, others were starting to play with their smartphones GPS function to find out about their current course and estimated arrival time. Unfortunately, due to the bad weather, the GPS worked really poorly and they only managed to get two GPS positions with a few minutes of time between them.

Given the coordinates the airplane was at at the times the GPS worked and the position of the runway threshold, can you find out how far away of the runway the plane is going to touch the ground, assuming it is moving continuously in the same direction since the first position was obtained?

The Amsterdam airport is at height 0, and you may assume the ground to be a flat plane (Amsterdam is in the Netherlands, after all).

Input

The input consists one line only, containing the coordinates of the first GPS position $0 \leq p1_x, p1_y, p1_h \leq 2000$, the second position $0 \leq p2_x, p2_y, p2_h \leq 2000$ and the position $0 \leq a_x, a_y \leq 2000$ of the runway threshold. $p1$ and $p2$ are guaranteed to differ in at least one component.

Output

Output a single number, how far away from the runway threshold the aircraft touches the ground. Your answer must be correct up to an absolute or relative tolerance of 10^{-6}. In case the aircraft is not going to land at any time soon, output the sentence: `WE LIKE TRAINS`

Sample Input 1

```
5 5 5 3 3 3 0 0
```

Sample Output 1

```
0
```

Sample Input 2

```
5 5 5 3 2 3 0 0
```

Sample Output 2

```
2.5
```

Sample Input 3

```
5 5 5 4 4 5 0 0
```

Sample Output 3

```
WE LIKE TRAINS
```

This page is intentionally left (almost) blank.