IDI Open Programming Contest March 30th, 2019

Problem Set

Jury and Problem Writers

Jean Niklas L'orange (Head Judge)
Torbjørn Morland
Karl Johan Heimark
Ruben Spaans

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.

Apostrophe Catastrophe

Problem ID: apostrophecatastrophe

Apostrophes and double quotes tend to get badly encoded by systems, causing them to print back things like `&#x27;` or `\"`. As a QA tester, Nova has seen lots of these issues. Today, she overheard that the Tweeper messaging app may have an issue with these symbols as well, and may encode them in strange ways. This was interesting, so she has written a Tweep to see if this is the case. Based on this Tweep, can you find out if Tweeper has an encoding issue, and if so, what the symbols are encoded into?

Input

The input is two strings separated by a line. The first string I represents the input characters that was sent in, and the second string O is how Tweeper presents the string. To avoid encoding troubles, apostrophes and double quotes have been replaced with + and −, respectively.

Output

The output should have n lines of output, one for each possible encoding + and − may have. Each line must have two strings a_i^+ and a_i^-, representing what a + and a − can be encoded as, respectively.

If the string is empty, print instead `<empty>`, and if the string can be anything, print `<any>`. If there are multiple answers, you can output them in any order.

If there is no valid answer, write `corrupted` instead.

Limits

- $1 \leq |I|, |O| \leq 280$
- I and O contain only the lowercase letters `a-z`, `0-9`, + and −

Sample Input 1

```
a+b-c
a-b+d-c
```

Sample Output 1

```
- +d-
```

Sample Input 2

```
knuth-morris-pratt
knuthmorrispratt
```

Sample Output 2

```
<any> <empty>
```

Sample Input 3

```
d+-trouble
doubletrouble
```

Sample Output 3

```
<empty> ouble
o uble
ou ble
oub le
oubl e
ouble <empty>
```

Bit 4 Bit
Problem ID: bit4bit

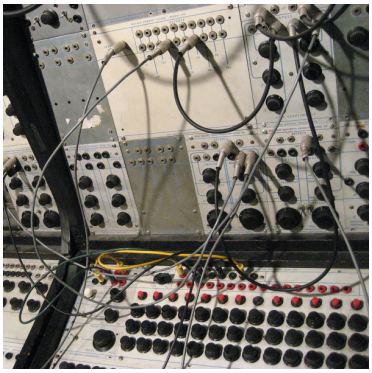

Photo by Bennett

The two robots ZF3 and XG2 are the hosts of the music show *Bit 4 Bit*. On the show, they play all kinds of songs, from the famous *Chips Ahoy Matey* to the lesser known ones, like *The Funky Algorithm*. Now, they only need some playlists for the next season!

ZF3 and XG2 start out with a single playlist and want to create P more. The first playlist p_0 contains a single song that lasts m minutes. To make more playlists, ZF3 and XG2 have two operations they can do:

1. Copy two existing playlists with the same number of songs and concatenate them.
2. Copy an existing playlist and replace one of the songs.

Since they usually don't have the time to play an entire playlist, they decide to pick a sublist instead of the entire thing. Now they wonder how long time it would take to play it.

Input

The first line contains three integers P, Q and m: The number of playlists to create, the number of sublists they want to check, and the length of the song in playlist p_0.

Then follow P lines describing how the ith playlist was created:

- `copy` p_i p_j – Copy playlist p_i, then copy p_j into the list afterwards
- `replace` p_i L_i T_i – Copy playlist p_i and replace song L_i with a song that lasts T_i minutes

Finally, Q queries on Q lines follow. Each line contains three integers p_i, A_i and B_i, which represent a query for the total play time of the playlist p_i from the song at index A_i, up to and including the song at index B_i.

Output

For each query, print out the time it will take to play all the songs in the sublist. Since the numbers may grow large, output the result modulo $1\,000\,000\,007$.

Limits

- $1 \leq P, Q \leq 200$
- $1 \leq m \leq 83$
- For `copy` operations, $|p_i| = |p_j|$
- For `replace` operations, $1 \leq T_i \leq 83$ and $0 \leq L_i < |p_i|$
- $0 \leq A_i \leq B_i < |p_i|$
- Playlist i will only copy playlists j where $j < i$

Sample Input 1	Sample Output 1
10 5 3 replace 0 0 5 copy 0 1 copy 1 0 copy 3 2 replace 4 2 4 replace 5 1 2 replace 4 3 7 copy 5 6 copy 7 5 copy 9 8 6 1 3 7 1 3 8 3 3 9 3 3 10 0 15	11 13 5 7 68

Coffee Date
Problem ID: coffeedate

Erika and Leah have not had a cup of coffee together for too long, and want to meet up today. Both of them like all the coffee shops in the city, so they decide to meet up at the coffee shop where they can see each other as soon as possible.

As they both live right by bus stops, they can take any of the buses that go by their home. They can also switch to another bus at a bus stop and can do so instantly, or wait until some other bus arrives.

The buses in the city are always on time, and all the routes have a new bus starting at some interval C_i *all day*. By pure chance, all the routes have a new bus starting right now.

As there is a coffee shop by every bus stop, Leah wants1 to know the earliest time they can see each other.

Photo by Vidar Nordli-Mathisen

Input

The first line contains four integers B, N, B_e, B_l: The number of bus stops, the number of bus routes, the bus stop by Erika's apartment, and the bus stop by Leah's apartment. Then follow $3N$ lines, three lines in a row for each bus stop.

Each bus route starts with a line with the integers C_i and S_i: The interval before a new bus starts, and the number of bus stops this route stops at, respectively. Then follows a line with S_i integers $s_{i,j}$, the number of each bus stop in order: $s_{i,0}$ is the start stop, $s_{i,1}$ is the next stop, and so on. After that comes a line with $S_i - 1$ integers $t_{i,j}$, the time it takes to go from bus stop $s_{i,j}$ to stop $s_{i,j+1}$.

Output

Output the earliest time Erika and Leah will see each other, assuming they both wait at the bus station right now. If there is no way they can meet, output '`NO COFFEE FOR YOU`'.

Limits

- $0 \leq B_e, B_l, s_{i,j} < B \leq 1\,000$
- $0 \leq N \leq 100$
- $1 \leq C_i, t_{ij} < 1\,000$
- $2 \leq S_i \leq B$
- A bus only arrives at a bus stop once

Sample Input 1	Sample Output 1
9 4 0 7 1 4 3 7 6 5 9 7 2 2 4 8 5 4 3 4 2 2 7 4 0 1 6 3 7 8 6 2 3 1 2 3 2 2	14

Delimiter Soup

Problem ID: delimitersoup

Photo by Judit Klein

Whenever a programmer starts to learn a Lisp, they think that there are too many parentheses in it. Sophia thinks there are too few, so she is making a programming language with only parentheses. To spice it up a bit, she is also adding square brackets (‘`[]`’) and curly braces (‘`{}`’) to the language.

Right now, she is struggling to make people use it for production code. Obviously, it has to be because of the bad error messages you get when you mess up the delimiters! Right now, you only get the error message ‘`syntax error`’ when you mess them up.

Any opening delimiter must be closed by the same type of delimiter: ‘`(`’ is closed with ‘`)`’, ‘`[`’ is closed by ‘`]`’, etc.

Sophia wants to improve the error message so that you at least get some help finding out where it all went wrong.

Input

The input consists of two lines. The first line contains an integer $|L|$, the length of the next line. The next line contains L, the program you want to validate.

Output

Output the character and the 0-indexed location of the first closing delimiter that does not match with the opening delimiter.

If there are no errors, or there are more opening delimiters than closing delimiters, print ‘`ok so far`’ instead.

Limits

- $1 \leq |L| \leq 200$
- L contains only the characters ‘`() [] {}`’ and spaces
- L does not start with a space character

Sample Input 1

```
8
([] [] ]
```

Sample Output 1

```
] 7
```

Sample Input 2

```
13
(([] [[]] ())
```

Sample Output 2

```
ok so far
```

Sample Input 3

```
21
[ { { () () () () } ]
```

Sample Output 3

```
] 20
```

Sample Input 4

```
27
[ { [[()]] (({})) } ] () {}
```

Sample Output 4

```
ok so far
```

Sample Input 5

```
19
[[]] () ) [] {{}} {
```

Sample Output 5

```
) 8
```

Excavator Expedition

Problem ID: excavatorexpedition

Little Nanouk is obsessed with excavators. In fact, he is so obsessed with them that he will throw a tantrum if he doesn't see too many during a road trip.

This poses a problem for his mother Amka, who wants to bring him to his grandparents today. To make the drive to his grandparents as smooth as possible, she has plotted out all the construction sites they could see on their way, along with the "boring" sites (according to Nanouk). Of course, Amka also wants to make some progress for each place they drive to. For that reason, the map she has plotted out is a directed acyclic graph.

Could you help Amka find the route to their grandparents which makes Nanouk as happy as possible?

Photo by Markus Spiske

Input

First, a single line with two integers V and E is given. Then, a single line with V characters is given, where V_i is '`X`' if location i is a construction site, and '`.`' if it is a boring site.

Then follow E lines. Every line contains two integers F_i and T_i, representing that it is possible to drive from F_i to T_i.

The location 0 and $V-1$ are Amka's home and Nanouk's grandparents' home, respectively.

Output

Output the happiness of Nanouk if Amka drives the route which maximises Nanouk's happiness. Nanouk's happiness is defined as the number of construction sites visited, minus the number of boring sites (except for Amka's and Nanouk's grandparents' house).

Limits

- $1 < V \leq 400\,000$
- $0 < E \leq 400\,000$
- $0 \leq F_i, T_i < V$
- For all $0 < i < V$, $V_i \in \{$'`.`', '`X`'$\}$
- For $i \in \{0, V-1\}$, $V_i =$ '`.`'
- There is always a path from Amka's house to Nanouk's grandparents.
- There are no cycles in the graph.

Sample Input 1	Sample Output 1
13 15 .X...X.X...X. 0 1 0 2 1 3 2 4 3 5 4 5 5 6 5 7 5 8 6 12 7 9 8 10 9 11 10 11 11 12	2

Forest Evolution
Problem ID: forestevolution

Ewan is a park ranger, and part of his duty is to observe how the ecosystem changes over time. He has just detected that a group of pine trees and aspens have started to share the same space. Before, there was a clear line between the species.

This isn't unusual, but Ewan is a bit curious about how this will evolve in the future. To track it, he wants to know how much of the area is covered by both species right now.

Photo by Patrick Hendry

Input

The input begins with a single line with two integers P and A: the number of pine trees and aspens, respectively. Then follow $P + A$ lines, one line for each tree. The first P of these lines contain locations for the pine trees, and the remaining A lines contain locations for the aspens.

Each tree is represented by two real numbers x_i and y_i, representing the location of the tree on a 2D map in metres.

Output

Output the area *covered* by both species in square metres. Your answer must have an absolute or relative error of at most 10^{-3}.

A point is *covered* by a species if it is at the edge of or inside a triangle made by three distinct trees of that species.

Limits

- $0 \leq P, A \leq 1\,000$
- $0.0 \leq x_i, y_i \leq 1\,000.0$
- $0.2 < |x_i - x_j| + |y_i - y_j|$ for all $0 \leq i < j < P + A$
- Real numbers in the input will have at most 8 digits after the decimal point

Sample Input 1

```
3 3
0.0 6.0
6.0 0.0
6.0 6.0
4.0 4.0
10.0 4.0
4.0 10.0
```

Sample Output 1

```
4.0
```

Sample Input 2

```
6 4
0.67 10.82
5.58 5.43
5.83 10.79
5.70 15.06
10.53 10.05
10.45 5.22
6.76 8.64
8.93 8.45
6.43 5.34
12.34 2.87
```

Sample Output 2

```
10.47478482125473
```

Game Suggestions

Problem ID: gamesuggestions

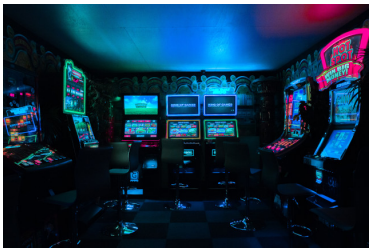

Photo by Carl Raw

Khadija is working on her new game, which will be a... well, she doesn't really know yet! She has made an extremely flexible game engine, and being a skilled developer, she can implement just about anything in it.

To find out what the game should contain, she has asked her friends for input. They have a lot of suggestions. So many, in fact, that to not overwhelm her with ideas, they have promised to only present at most one suggestion each.

However, they all want to give Khadija as many different suggestions as possible, so they have gathered and found a set of suggestions such that:

- They maximise the total number of suggestions they tell to Khadija
- Every person suggests at most one of their own suggestions

When Khadija has received all the suggestions, she assigns them to one or more categories.

To get some variation into the game, she limits the number of suggestions from one category. She then picks as many suggestions as possible, given the category limits (A suggestion only counts towards one category).

While she's busy implementing all these suggestions, her friends wonder how many suggestions she could've ended up implementing in theory.

Input

The input begins with a single line with three integers F, S and C: The number of friends, distinct suggestions and distinct categories, respectively.

Next follow F lines, one line for each friend: The line contains Fn_i space separated suggestion names.

Then, C lines follow, one for each category. The line starts with an integer Cn_i, the maximum number of suggestions Khadija wants to include that are assigned to this category, followed by Sn_i space separated suggestion names which are part of this category.

Output

Output the maximum number of suggestions Khadija could possibly end up implementing.

Limits

- $0 < F, S, C, Fn_i \leq 200$
- $0 < Cn_i \leq Sn_i \leq 100$
- All suggestion names are unique, and consist of between 1 and 20 characters in the range `a-z` and `0-9`.

Sample Input 1

```
7 9 4
ducks 2d
guns 3d swords
3d swords
3d swords wolves
swords 2d
zombies 2d vampires demons
2d swords
1 2d 3d
1 swords guns
1 wolves ducks
4 wolves zombies vampires demons
```

Sample Output 1

```
5
```

Helpful Currents

Problem ID: helpfulcurrents

Photo by Payton Ferris

Lysias loves his full-rigged ship and takes it out to his island castle as often as possible. On his way back to the castle one day, the engines capable of turning the sails failed. As it is a full-rigged ship, he and his crew are now unable to turn the sails in any direction.

But, as luck would have it, there is a south wind which is blowing them in the right direction... at least approximately. Since they may not get to the castle by wind only, Lysias believes he can use the ocean's currents to move the ship either east or west temporarily by retracting and lowering the ship's sails. In that way, the ship can move in two potential directions:

- Move north by lowering the sails
- Retract the sails and move in the direction of the current they're on top of (east/west)

Lysias has dug up an old map of all the currents in this part of the ocean. As he is fond of mathematics, not only does Lysias wonder if it's possible to get home without fixing the sails; he also wonders how many different routes they can take to get home. Can you help him?

Input

The first line has three integers: Y and X and x_{init}, representing the number of rows and columns of the map, and which column the ship is currently placed at. The ship always starts on the bottom row.

Then follow Y rows, each with X characters each. All characters $C_{x,y}$ is one of '~', '#', '@', '>' and '<'. '~' represents open sea without any currents, '#' is impassable shallow waters, and '>' and '<' are currents moving the boat to the right and left respectively. '@' represents Lysias' castle.

Output

Output all the different distinct paths the ship could take to get back to the castle. Since there may be very many different ways to get back, output the answer modulo $1\,000\,003$.

If there are no ways to get to the castle, output "`begin repairs`".

Limits

- $0 < Y \leq 300$
- $0 \leq x_{init} < X \leq 50\,000$
- If $C_{x,y} =$ '>', then $x + 1 < X$ and $C_{x+1,y} \notin \{$'<', '#'$\}$
- If $C_{x,y} =$ '<', then $0 \leq x - 1$ and $C_{x-1,y} \notin \{$'>', '#'$\}$
- There is exactly one '@' tile on the map
- The boat will not start on a '#' tile

Sample Input 1	Sample Output 1
2 2 0 >@ >~	2

Sample Input 2

```
3 5 1
>>@<<
>~#~<
>>>>~
```

Sample Output 2

```
4
```

Sample Input 3

```
3 4 0
>~@~
~<#~
>>>~
```

Sample Output 3

```
begin repairs
```

if-then-else
Problem ID: ifthenelse

Saladin loves to make computers and languages. This time, he has made a CPU that works on 12-bit words, and created his own language called Salang on top of it. The language itself works well, except for one weird bug. To fix the problem, he has designed a decompiler: A program that takes the machine code and creates an intermediate representation of the program.

The intermediate representation is in a small language that can only do assignments, integer addition and multiplication, if statements and printing. The integer operations silently overflow and wrap around, and the representation has no scoping (a variable `@var` has the same memory address everywhere). It uses the following EBNF syntax:

```
<letter> := 'a' | 'b' ... 'y' | 'z'
<var> := '@' letter { letter }
<uint> := ? base 10 of an unsigned int lower than 2^12 ?
<val> := <var> | <uint>
<op> := '+' | '*'
<cmp> := '==' | '<=' | '<'
<assign> := <var> '=' <val> [ <op> <val> ] '\n'
<if> := 'if' <val> <cmp> <val> 'then' '\n' { <statement> }
        [ 'else' '\n' { <statement> } ]
        'endif' '\n'
<print> := 'print' <val>
<statement> := <assign> | <if> | <print>
<program> := { <statement> }
```

where all values are separated by exactly one space, unless if they begin on a new line.

But the decompiler has forgotten to output the first line! Although all variables are initialised to 0 before the program starts, the program always begins by assigning the magic variable `@a` to a fixed number. Before Saladin can start with the debugging, he needs to know the initial value of `@a`.

Input

The input begins with a single line with two integers: I and O. Then comes I lines, the program itself. Finally comes O lines, the numbers this program printed when it ran.

Output

Output the value `@a` has at the start of the program. If there are multiple valid answers, output the lowest one. If there are no valid answers, output "`no solution`".

Limits

- $0 \leq O < I \leq 100$
- The program matches the EBNF above
- The length of a line will always be less than 80 characters

Sample Input 1	Sample Output 1

12 2
@b = @a + @a
@c = @a * @a
if @c <= @b then
@dx = @b + 3996
if @dx < @a then
print @c
else
endif
endif
@a = @b * @dx
@a = @a + @c
print @a
129
4029
	65

Job Expenses

Problem ID: jobexpenses

At ACME Corp, the seller Robin is selling software like hotcakes. To sell the software, he arranges meetings all over the country. Of course, he always needs to travel there, and sometimes even stay the night. Other times, he decides that a hot dog on the way to a meeting counts as accommodation.

Because Robin is very busy, he hasn't been able to file his job expenses this month. Eret, the accountant, told him that he could fix it for him, as long as he got the price of all the expenses along with the receipts. Robin did this but misheard him: He has provided a list of all expenses *and* incomes he has brought in last month, with incomes as positive integers and expenses as negative integers.

Can you help Eret find out how much expenses Robin has paid for the last month?

Input

The input consist of two lines: One line with the integer N, and one line with N integers k_i, representing the numbers Robin has written down.

Output

Output the sum of all the expenses Robin has paid for the last month.

Limits

- $0 \leq N \leq 20\,000$
- $-50\,000 \leq k_i \leq 500\,000$ and $k_i \neq 0$

Sample Input 1

```
3
1 -2 3
```

Sample Output 1

```
2
```

Sample Input 2

```
5
129 44 394 253 147
```

Sample Output 2

```
0
```

Sample Input 3

```
10
-100 40000 -6500 -230 -18 34500 -450 13000 -100 5000
```

Sample Output 3

```
7398
```