German Collegiate Programming Contest 2023

June 17th

Problems

This page is intentionally left (almost) blank.

Problem A: Adolescent Architecture 2
Time limit: 2 seconds

Three years ago, you helped little Peter stack his toy blocks into a tower. Since then, he has extended his collection of toy blocks, which now features the following base shapes:

- `circle` a – a circle of radius a;
- `square` a – a square with side length a;
- `triangle` a – an equilateral triangle with side length a.

Here, a may be any positive integer. The top shapes of each block are the same as their bottom shapes, so the blocks are cuboids, cylinders, and triangular prisms, respectively. Peter has an infinite supply of blocks of each shape and size.

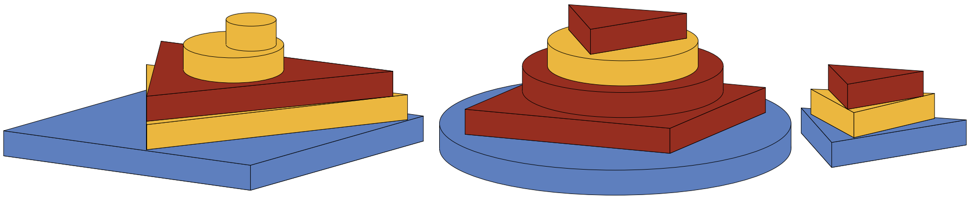


Figure A.1: A game in progress.

Peter and his friend Amy are playing a two-player game, where the blocks need to be stacked on top of each other. Initially, some blocks are already placed on the floor. In each move, the current player must take a toy block from the infinite supply and put it on top of one of the existing stacks of blocks. The block may be rotated around its vertical axis before placing it. The outline of the new block must be strictly within the outline of the old block; the outlines are not allowed to touch. The first player who is unable to make a move loses the game.

Given the initial configuration, determine the number of winning moves for the first player.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 1000$), the number of initial stacks.
- n lines, each with a string s (s is one of "`circle`", "`square`" or "`triangle`") and an integer a ($1 \leq a \leq 10^9$), giving the topmost blocks of the initial stacks as described above.

Output

Output the number of winning moves for the first player.

Sample Input 1	Sample Output 1
2 circle 2 triangle 2	2

Sample Input 2	Sample Output 2
2 circle 1 circle 2	3

Sample Input 3

```
5
circle 123
triangle 456
square 789
square 789
triangle 555
```

Sample Output 3

```
7
```

Sample Input 4

```
3
circle 299303201
square 79724391
triangle 437068198
```

Sample Output 4

```
3
```

Sample Input 5

```
3
square 539715887
circle 518408351
triangle 348712924
```

Sample Output 5

```
0
```

Notes

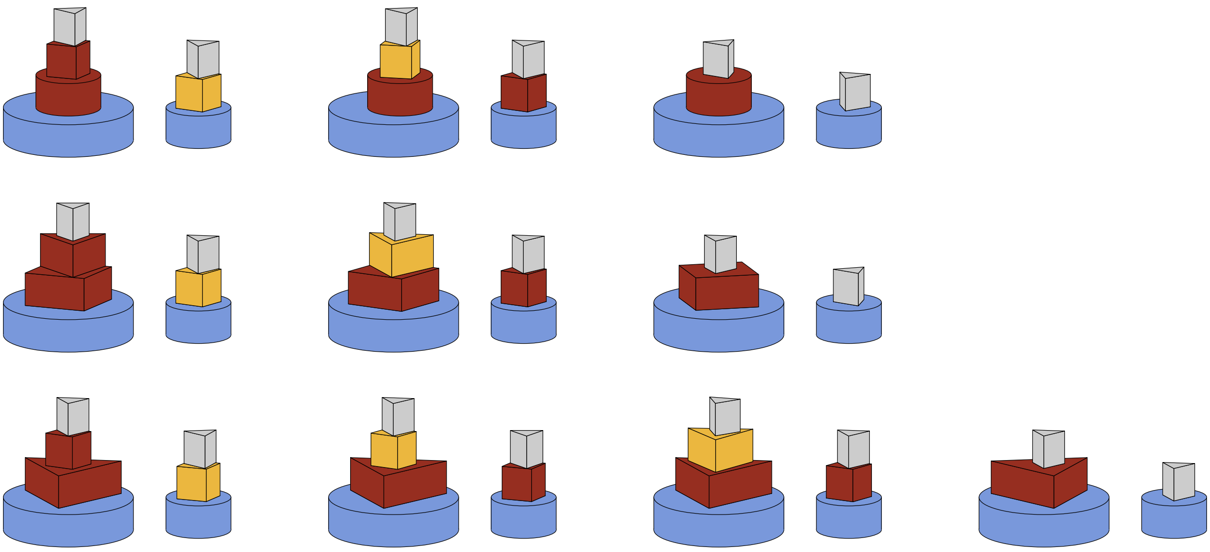


Figure A.2: Illustration of Sample Input 2, showing all possible end configurations of the game when Peter went first and played optimally to win. The blue blocks are the initial configuration. Peter needs to put one of `circle 1`, `square 2` or `triangle 3` on top of `circle 2` in order to win. Each of these options corresponds to one row of the figure. Blocks placed by Peter are coloured in red, and blocks placed by Amy are coloured in yellow. As the last two blocks are always of type `triangle 1`, they are shown in grey. If, for instance, Peter first puts `circle 1` (as depicted in the first row), then Peter can win by mirroring the following moves by Amy.

Problem B: Balloon Darts

Time limit: 1 second

Popping balloons as an amusement park attraction. Photo by blende12, Pixabay

As you may know, you get a colourful balloon for each problem you solve in an ICPC contest. You were quite successful in your last contest and now you own a remarkable collection of n balloons. The obvious thing to do with these balloons is to pop them all using darts. However, you only have three darts.

The balloons are modelled as points in the plane with fixed locations. For each dart you choose from where and in which direction to throw it. The dart travels in a straight line, popping all balloons in its way.

As you practised a lot during the last years, you can throw a dart precisely in any direction and it will fly infinitely far. Thus, if anyone can pop all the balloons, it is you. However, before the fun begins, you first need to determine if you can pop all balloons using at most three darts.

Input

The input consists of:

- One line containing an integer n ($1 \leq n \leq 10^4$), the number of balloons.
- n lines, each containing two integers x and y ($|x|, |y| \leq 10^9$), the coordinates of a balloon.

It is guaranteed that no two balloons are at the same location.

Output

Output "`possible`" if three darts are sufficient to pop all balloons and "`impossible`" otherwise.

Sample Input 1	Sample Output 1
6 0 0 1 1 2 4 3 9 4 16 5 25	possible

Sample Input 2	Sample Output 2
7 0 0 1 1 2 4 3 9 4 16 5 25 6 36	impossible

Sample Input 3

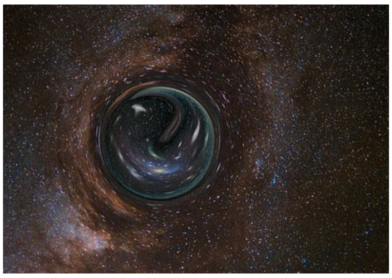
```
7
-1 -1
0 0
1 1
2 4
3 9
4 16
5 25
```

Sample Output 3

```
possible
```

Problem C: Cosmic Commute

Time limit: 3 seconds



A wormhole above Gallifrey, mau_king

A long time ago, in a galaxy far, far away, the InterCosmic Passage Company (ICPC) operates a complex railway system using *light trains*. Each planet has exactly one train station and each light train connects two distinct planets of the galaxy, going back and forth between them. Just recently, the InterCosmic Passage Company established a teleportation system, which is now in its testing phase. Some train stations are now extended by a *wormhole*. All wormholes are connected to each other, and it is possible to teleport from one wormhole to another instantaneously. To not overload the new system, each citizen of the galaxy is only allowed to teleport at most once a day.

Charlie lives on planet Gallifrey and works on planet Sontar. It is her first day of work, and she is already terribly late because her stupid alarm clock did not go off. On top of that, the new teleportation system is malfunctioning today of all days, and the destination wormhole cannot be chosen. Instead, after entering a wormhole, one is teleported to a wormhole that is chosen uniformly at random among all other wormholes. (It is impossible to be at the same train station after teleportation.)

Despite all her bad luck, Charlie is dead set on getting to work on time. Since all light trains are very slow, she wants to take as few light trains as possible. What is the expected minimum number of light trains she has to take to get to work if she can use the (malfunctioning) teleportation system at most once?

Input

The input consists of:

- One line with integers n, m, k ($2 \leq n \leq 2 \cdot 10^5$, $n - 1 \leq m \leq 10^6, 2 \leq k \leq n$), the number of planets in the galaxy, light trains and wormholes. Planet 1 is Charlie's home planet Gallifrey, and planet n is Sontar, where Charlie works.
- One line containing k distinct integers, the planets whose train stations each have a wormhole (in addition to the light trains).
- m lines, each containing two integers a and b ($1 \leq a, b \leq n$ and $a \neq b$), describing a light train between the planets a and b. It is guaranteed that all light trains are pairwise disjoint.

It is guaranteed that it is possible to travel from any planet to any other planet of the galaxy using only light trains.

Output

Output a single reduced fraction, the expected minimum number of light trains Charlie has to take to get to work if she can use the (malfunctioning) teleportation system at most once. Output the fraction as "a/b", where a is the numerator and b is the denominator.

Sample Input 1

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

Sample Output 1

```
5/2
```

Sample Input 2

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

Sample Output 2

```
2/1
```

Problem D: DnD Dice

Time limit: 1 second

The five standard dice, Ramona Trusheim

In *Dungeons & Dragons* (DnD) and many other role playing games, many actions are determined by dice rolls, and it is also quite common to use dice with different numbers of sides. The most common dice are those based on the five Platonic solids, the tetrahedron, cube, octahedron, dodecahedron and icosahedron, with 4, 6, 8, 12 and 20 sides, respectively. In DnD terminology, these dice are usually called d4, d6, d8, d12 and d20.

As a dungeon master, you are currently designing a campaign for your group of players. In the final battle of this campaign, the players need to roll a combination of multiple dice with varying numbers of sides, and the action of the enemy is determined by the sum of the numbers on the dice that were rolled. For balancing purposes, you want to sort these sums based on how likely they are to occur, so that you can assign appropriate events to each of them.

Given the number of dice of each type, and assuming the sides of each die are numbered from 1 to the number of sides, find all possible sums of dice rolls and output them sorted by non-increasing probability.

Input

The input consists of:

- One line with five integers t, c, o, d and i, ($0 \leq t, c, o, d, i \leq 10$), giving the number of tetrahedra, cubes, octahedra, dodecahedra and icosahedra among the dice that are rolled. There is always at least one die, that is $t + c + o + d + i \geq 1$.

Output

Output all possible sums, ordered from most likely to least likely. If two sums occur with the same probability, then those sums may be printed in any relative order.

Sample Input 1

```
1 1 1 0 0
```

Sample Output 1

```
11 10 9 12 8 13 14 7 15 6 5 16 17 4 18 3
```

Sample Input 2

```
2 0 0 1 0
```

Sample Output 2

```
9 14 12 11 10 13 15 8 16 7 6 17 5 18 4 19 3 20
```

Sample Input 3

```
0 0 0 0 1
```

Sample Output 3

```
10 3 9 8 14 13 11 7 6 12 20 4 15 5 16 1 2 19 17 18
```

Problem E: Eszett

Time limit: 1 second

```
jury@GCPC:~/gcpc23$ python3
Python 3.11.2 on linux
Type "help" for more information.
>>> def test(s):
...         print(s.upper())
...
>>> test('python ist großartig!')
PYTHON IST GROSSARTIG!
>>> █
```

A demonstration of the `upper` function in Python

For those trying to learn German, the letter 'ß', called *Eszett* or *sharp S*, is usually a source of great confusion. This letter is unique to the German language and it looks similar to a 'b' but is pronounced like an 's'.

Adding to the confusion is the fact that, until a few years ago, only a lowercase version of 'ß' existed in standard German orthography. Wherever an uppercase 'ß' was needed, for example in legal documents and shop signs, it was (and usually still is) replaced by capital double letters 'SS'. In 2017, the capital 'ẞ' was officially introduced into the German language and may now be used in those scenarios, instead.

Other than being confusing for foreigners, the practice of replacing 'ß' with 'SS' also introduces some ambiguity because a given uppercase word featuring one or more occurrences of 'SS' may correspond to multiple different lowercase words, depending on whether each 'SS' is a capitalized 'ß' or 'ss'.

Given one uppercase word, find all the lowercase words that it could be derived from. As the letter 'ß' is not part of the ASCII range, please write an uppercase 'B', instead.

Input

The input consists of:

- One line with a string s ($1 \le |s| \le 20$) consisting of uppercase letters.

It is guaranteed that the letter `S` occurs at most three times in s. Note that s need not be an actual German word.

Output

Output all the possible lowercase strings corresponding to s. Any order will be accepted, but each string must occur exactly once.

Sample Input 1	Sample Output 1
`AUFREISSEN`	`aufreissen` `aufreiBen`

Sample Input 2	Sample Output 2

`MASSSTAB`	`massstab`
	`maBstab`
	`masBtab`

Sample Input 3	Sample Output 3
`EINDEUTIG`	`eindeutig`

Sample Input 4	Sample Output 4
`S`	`s`

Sample Input 5	Sample Output 5
`STRASSE`	`strasse` `straBe`

Problem F: Freestyle Masonry

Time limit: 2 seconds

Fred got a simple task, he just has to build a $w \times h$ wall. To make this even easier, he was provided with enough 2×1 bricks and also a few 1×1 bricks to complete the wall. Knowing that this task should not be too hard, Fred went to work and started building the wall without thinking too much about the design. Only when he ran out of 1×1 bricks, Fred noticed that this might have been a bad idea...

An interesting brick layout, photo by Bobo Boom

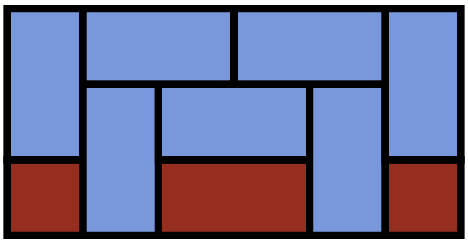


Figure F.1: Visualization of Sample Input 2. The red bricks have already been placed by Fred. The blue bricks still need to be placed to complete the wall (the only possible design in this case).

Maybe he should have made a plan before starting to build the wall, but now it is too late. Fred only has a bunch of 2×1 bricks left and wants to finish the wall. Can he still complete it with the remaining 2×1 bricks? Note that the wall to be built should have a width of exactly w units and a height of exactly h units.

Input

The input consists of:

- One line with two integers w and h ($1 \leq w \leq 2 \cdot 10^5$, $1 \leq h \leq 10^6$), the width and height of the wall Fred wants to build.
- One line with w integers $h_1, \ldots, h_n$ ($0 \leq h_i \leq 10^6$), where h_i is the current height of the wall at position i.

Output

Output "`possible`" if Fred can complete his wall and "`impossible`" otherwise.

Sample Input 1	Sample Output 1
3 3 0 0 1	possible

Sample Input 2	Sample Output 2
6 3 1 0 1 1 0 1	possible

Sample Input 3	Sample Output 3
6 2 1 0 1 1 0 1	impossible

Sample Input 4	Sample Output 4
5 2 1 2 3 2 2	impossible

Problem G: German Conference for Public Counting

Time limit: 1 second

Greta loves counting. She practises it every day of the year. Depending on the season, she counts falling leaves, raindrops, snowflakes, or even growing leaves. However, there is one event in summer which tops everything else: the German Conference for Public Counting (GCPC).

At this event, Greta meets counting enthusiasts from all over the country for one week of counting and counting and counting… Together they participate in the Glamorous Competitive Public Counting and the Great Chaotic Public Counting. At the end of the week they all try to win the Golden Cup of Public Counting. Her favourite is the Gently Calming Public Counting where the crowd counts in silence, trying to harmoniously synchronise to reach the target number at precisely the same moment.

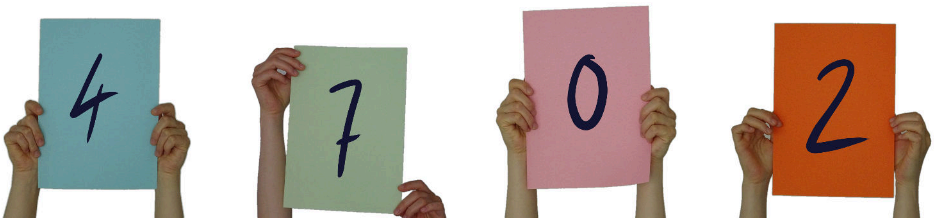


Figure G.1: People holding up signs for the countdown.

To increase the tension and to prepare for the Gently Calming Public Counting, the organizers of GCPC plan to start with a silent countdown, where the people on the stage will at any time display the current number by holding up signs with its digits. On every sign, there is exactly one decimal digit. Numbers greater than 9 are displayed by holding up several signs next to each other. Each number is shown using the least possible number of signs; there is no left padding with zeroes. This way, the people on the stage display numbers $n, n-1, n-2, \ldots$ until they finally display 0. Since the GCPC will take place soon, the organizers want to finish their preparations quickly. How many signs do they need to prepare at least so that they can display the entire countdown from n to 0?

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 10^9$), the starting number of the countdown.

Output

Output the minimum number of signs required to display every number of the countdown.

Sample Input 1	Sample Output 1
5	6

Sample Input 2	Sample Output 2
20	11

Sample Input 3	Sample Output 3
44	14

Sample Input 4	Sample Output 4
271828182	82

Sample Input 5	Sample Output 5
314159265	82

Notes

In the first sample case, the organizers need one sign each with the digits 0 to 5, for a total of 6 signs. In the second sample case, they need one sign with each digit other than 1, and two signs with a 1, for a total of $9 + 2 = 11$ signs.

Problem H: Highway Combinatorics

Time limit: 3 seconds

Jam caused by parking bus, Nevermind2

You are the new minister of transport in Berland. Recently, you allowed free parking on a two lane road segment of 200 metres length. Since then, that road segment has constantly been jammed by parked cars due to some genius drivers who had the idea to park their cars across both lanes...

However, this is not your concern. You are more interested in parking some of your own cars on the road segment while it is empty. More specifically, you want to park some of your cars in such a way that the number of different ways to fill the remaining empty space with cars is congruent to your lucky number n modulo $10^9 + 7$.

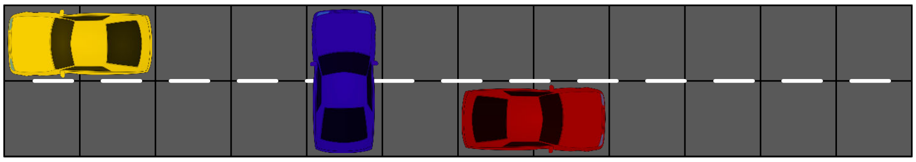


Figure H.1: Visualization of Sample Output 1.

Each car has a size of 1×2 metres, each of the two lanes is 1 metre wide and 200 metres long. You own more than 200 cars which you could park on the road segment.

Input

The input consists of:

- One line with one integer n ($0 \leq n < 10^9 + 7$), the desired number of possible ways to fill the road modulo $10^9 + 7$.

It can be proven that at least one valid solution exists for each possible value of n.

Output

Output the state of the two lanes in two lines. Print "#" for an occupied spot and "." for an empty spot. Note that the two lines should have the same length of at least 1 metre and at most 200 metres, and the occupied spots must correspond to some parked cars. If your solution uses a road segment shorter than 200 metres, the remaining part of the road segment is assumed to be blocked by parked cars.

Sample Input 1	Sample Output 1
10	##..#....... ....#.##....

Sample Input 2	Sample Output 2
27	...##....... ....##...

This page is intentionally left (almost) blank.

Problem I: Investigating Frog Behaviour on Lily Pad Patterns

Time limit: 3 seconds

A frog in a pond.

Recently, the biologist Ina discovered a new frog species on the lily pads of a pond. She observed the frogs for a while and found them to be very conscious about their personal space because they avoided sharing a lily pad with other frogs. Also, they seemed quite lazy as they did not move often, and if they did, they always jumped to the nearest empty lily pad.

To confirm her hypotheses about the frogs' movement pattern, Ina set up a large number of lily pads in a pool in her laboratory, arranged in a straight line. Since the frogs were attracted to light, she was able to simplify the test setup further by placing a bright light at one end of that line. This way, the frogs would always jump in one direction (towards the light).

Of course, Ina could now place some frogs on the lily pads and sit there all day watching the frogs jump around. But as the frogs move so rarely, it would take ages to gather a sufficient amount of data.

She therefore attached to each frog a tiny device that could log all jumps of that frog. This way, she could put the frogs on the lily pads, leave them alone for a few hours and come back later to collect the data. Unfortunately, the devices had to be so tiny that there was no space for a position tracking system; instead, the devices could only record the times of the jumps.

But if the movement pattern of the frogs is as restricted as Ina thinks, surely the individual movements of the frog can be reconstructed only from the initial positions and the recorded jump time stamps?

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 2 \cdot 10^5$), the number of frogs.
- One line with n integers $x_1, \ldots, x_n$ ($1 \leq x_i \leq 10^6$), the number of the lily pad on which the ith frog initially sits. The lily pads are numbered consecutively, starting at 1. It is guaranteed that the initial positions are strictly increasing, i.e. $x_1 < x_2 < \cdots < x_n$.
- One line with an integer q ($1 \leq q \leq 2 \cdot 10^5$), the number of jumps recorded.
- q lines, each containing an integer i ($1 \leq i \leq n$), indicating that the ith frog jumped. The jumps are given in chronological order and you may assume that a jumping frog lands before the next jump begins. The frogs always jump to the nearest empty lily pad with a larger number, and you may assume that such a lily pad always exists.

Output

For each jump, output the number of the lily pad the frog lands on.

Sample Input 1	Sample Output 1
5 1 2 3 5 7 3 1 2 4	4 6 8

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

Notes

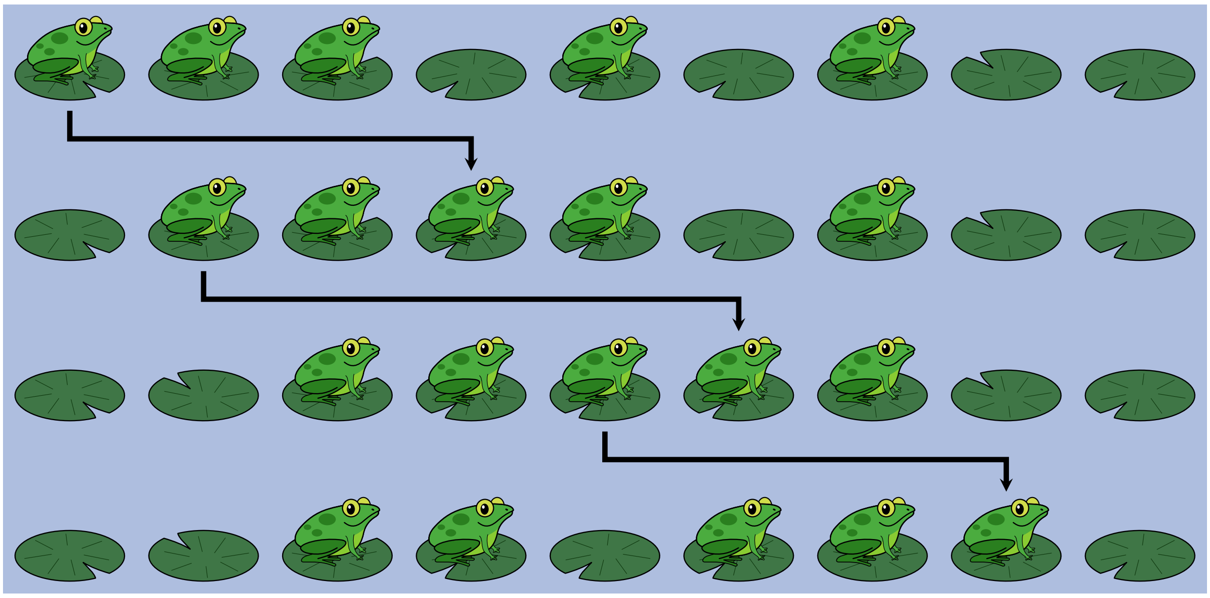
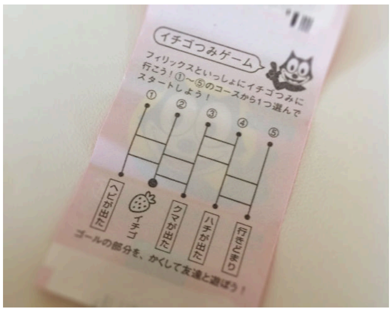


Figure I.1: Illustration of the first sample case. The lily pads are numbered from left to right, starting at 1.

Problem J: Japanese Lottery

Time limit: 4 seconds

Amida-kuji is a lottery popular in Japan, which can be used to assign w prizes to w people. The game consists of w vertical lines, called *legs*, and some horizontal bars that connect adjacent legs. The tops of the legs are the starting positions of the w people, and the prizes are at the bottom of the legs. To determine the prize of the ith person, one has to move down on the ith leg, starting at the top, and switch the leg whenever a horizontal bar is encountered. You can see such a game and how to trace a path in Figure J.1.



Strawberry picking game, photo by Nanao Wagatsuma

You want to manipulate the lottery in such a way that the ith person gets the ith prize, for every i, by removing some horizontal bars. Since you do not want to get caught, you want to remove as few bars as possible.

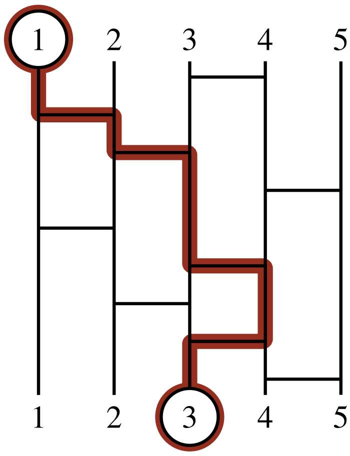


Figure J.1: Visualization of an Amida-kuji game. The first person is connected to the third prize. This is also Sample Input 2 after all connections are added and before any connection is removed. To connect the ith person to the ith prize, it suffices to remove both horizontal bars between legs 2 and 3 and the topmost horizontal bar between legs 3 and 4. This is the only minimal solution.

For this problem, the initial game configuration has no horizontal bars. Then, horizontal bars are added one by one or are removed again. After each change, you want to know the minimum number of horizontal bars that need to be removed such that the ith prize is assigned to the ith person for each i. Note that this is always possible by removing all horizontal bars.

Input

The input consists of:

- One line with three integers w, h and q ($2 \leq w \leq 20$, $1 \leq h, q \leq 2 \cdot 10^5$), the number of legs, the height of the legs, and the number of changes.
- q lines, each containing three integers y, x_1 and x_2 ($1 \leq y \leq h$, $1 \leq x_1, x_2 \leq w$), describing a change where a horizontal bar is added or removed at height y between legs x_1 and x_2. If there is already a horizontal bar at this position, it will be removed. Otherwise the bar will be added. It is guaranteed that the two legs are adjacent, i.e. $|x_1 - x_2| = 1$.

It is guaranteed that all horizontal bars have different heights at every moment.

Output

After each change, output a single integer, the minimum number of horizontal bars that need to be removed in the game with the currently existing bars such that the ith prize is assigned to the ith person for each i.

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

Sample Input 2	Sample Output 2
5 9 12 1 3 4 2 1 2 3 2 3 4 4 5 5 2 1 6 4 3 7 2 3 8 4 3 9 4 5 6 4 3 7 2 3 1 3 4	1 2 3 4 3 2 3 4 3 4 3 2

Problem K: Kaldorian Knights

Time limit: 2 seconds

Painting of a medieval tournament, Codex Manesse.

The king of Kaldoria traditionally celebrates his birthday by inviting the knights of his realm to a big jousting tournament, and every noble house participates by sending their best knights to win fame and glory. At the end of the tournament, the king does not only choose a winner but ranks all n knights from worst to best.

The number of knights belonging to house i is denoted by k_i. Each knight serves at most one house. There can be knights who do not serve any house. The houses are ordered by their influence in the kingdom (with the first one being the most influential). If the k_1 knights of the most powerful house take the last k_1 places in the tournament, the house will incite a revolt against king and crown. The members of the second most influential house are not that powerful. Even if all its k_2 knights end up at the bottom of the ranking, it would be considered a strong provocation but the house would not be able to start a revolt. However, if the last $k_1 + k_2$ places are taken by all the knights of the two most influential houses combined, then the two houses will unite and start fighting the king. More generally, if the knights of the ℓ most powerful houses occupy the last $k_1 + k_2 + \cdots + k_\ell$ places in the tournament, they will unite and incite a revolt.

Of course, a revolt has to be avoided at all cost. Knowing that the king often chooses the ranking spontaneously and without too much consideration, the chief mathematician of the crown has been tasked with analysing how many rankings will *not* lead to a revolt.

Input

The input consists of:

- One line with integers n ($1 \le n \le 10^6$) and h ($0 \le h \le 5000$), the number of knights and the number of houses.
- h lines, with the ith line containing an integer k_i ($1 \le k_i \le n$), denoting the number of knights from house i. Note that every house is represented by at least one knight.

It holds that $\sum_{i=1}^{h} k_i \le n$.

Output

Output the number of rankings that do not lead to a revolt. As this number can be quite large, it should be output modulo $10^9 + 7$.

Sample Input 1	Sample Output 1
3 0	6

Sample Input 2	Sample Output 2
4 1 3	18

Sample Input 3	Sample Output 3
4 2 2 1	16

Problem L: Loop Invariant

Time limit: 1 second

A circular piece of parchment.

Luna, a historian, was exploring the archives of an old monastery when she stumbled on a mysterious scroll of parchment. On it were only two types of symbols: '`(`' and '`)`'. Soon she noticed that the sequence of symbols satisfies an interesting property: It can be constructed by repeatedly inserting '`()`' at some position into an initially empty sequence. Historians call such sequences *balanced*. Figure L.1 gives an example of a balanced sequence.

`()` $\rightarrow$ `()()` $\rightarrow$ `()()()` $\rightarrow$ `()(())()`

Figure L.1: Sample Input 2 derived by successively inserting '`()`' into an initially empty sequence.

The chief librarian of the monastery recently told Luna that some of the more elitist monks in the region had a habit of writing on circular pieces of parchment. In their minds, anyone incapable of immediately telling where the text on such a scroll started was also unworthy of knowing its content. Consequently, Luna quickly inspected the edges of her parchment strip. And sure enough, the edges at the left and right end of the parchment strip fit together perfectly, indicating that the parchment once actually was circular. While holding the left and right edges together and looking at the now circular parchment, she wonders whether the balanced sequence starting at the tear is the only such sequence that could have resulted from tearing the parchment apart. After all, it makes little sense trying to decrypt a text when you do not even know where it starts.

Input

The input consists of:

- One line with a balanced sequence s ($2 \leq |s| \leq 10^6$), the sequence on Luna's strip of parchment.

Output

Output "`no`" if there is no way to obtain a different balanced sequence by cutting the circular sequence, otherwise give any such sequence.

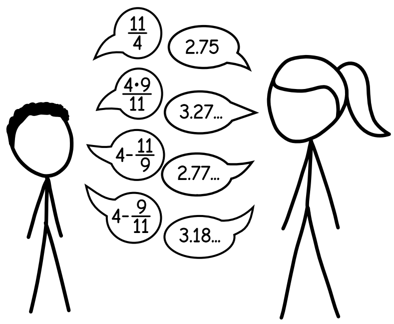
Sample Input 1	Sample Output 1
`(()())(()())`	`no`

Sample Input 2	Sample Output 2
`()(())()`	`(())()()`

This page is intentionally left (almost) blank.

Problem M: Mischievous Math

Time limit: 1 second



Max trying to solve the third sample.

Max enjoys playing number games, whether it involves finding a combination that leads to a given result or discovering all possible results for some given integers. The problem is that Max is only 10 and has limited mathematical knowledge, which restricts the possibilities for these games. Luckily, in today's maths class, Max learned the concept of parentheses and their effects on calculations. He realises that incorporating parentheses into his number games could make them much more interesting. After getting home from school, he asked his sister Nina to play a variant of his favourite number game with him, using parentheses.

In this new game, Max first tells her a number d. Nina then tells him three numbers a, b and c. Now, Max needs to find an arithmetic expression using addition, subtraction, multiplication and division, using each of these three numbers (a, b and c) at most once, such that the result is equal to d. The numbers a, b, c and d all have to be distinct, and Max is allowed to use parentheses as well.

For instance, for $a = 5$, $b = 8$, $c = 17$ and $d = 96$ a possible solution would be $(17-5)\times 8 = 96$, and for $a = 3$, $b = 7$, $c = 84$ and $d = 12$ a possible solution would be $84 \div 7 = 12$, without using the 3.

Nina is quickly annoyed by this game. She would rather spend the afternoon with her friends instead of playing games with her little brother. Therefore, she wants to give him a task that occupies him for as long as possible. Help her to find three numbers a, b and c such that it is impossible for Max to come up with a solution.

Input

The input consists of:

- One line with an integer d ($1 \leq d \leq 100$), Max's chosen number.

Output

Output three numbers a, b and c ($1 \leq a, b, c \leq 100$) such that the numbers a, b, c and d are pairwise distinct and there is no solution to the number game.

Sample Input 1	Sample Output 1
`5`	`1 20 30`

Sample Input 2	Sample Output 2
`100`	`33 55 77`

Sample Input 3	Sample Output 3
`3`	`11 9 4`

This page is intentionally left (almost) blank.