ICPC SOUTH PACIFIC DIVISIONALS

OCTOBER 16, 2021

Contest Problems

This contest contains thirteen problems over 28 pages. Good luck.

For problems that state "*Your answer should have an absolute or relative error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $|x - y| < 10^{-9}$ or $\frac{|x-y|}{|y|} < 10^{-9}$, then your answer will be considered correct.

Definition 1

For problems that ask for a result modulo m:
If the correct answer to the problem is the integer b, then you should display the unique value a such that:

- $0 \leq a < m$
  and
- $(a - b)$ is a multiple of m.

Definition 2

A string $s_1 s_2 \cdots s_n$ is lexicographically smaller than $t_1 t_2 \cdots t_\ell$ if

- there exists $k \leq \min(n, \ell)$ such that $s_i = t_i$ for all $1 \leq i < k$ and $s_k < t_k$
  or
- $s_i = t_i$ for all $1 \leq i \leq \min(n, \ell)$ and $n < \ell$.

Definition 3

- Uppercase letters are the uppercase English letters $(A, B, \ldots, Z)$.
- Lowercase letters are the lowercase English letters $(a, b, \ldots, z)$.

Definition 4

Unless otherwise specified, the distance between two points (x_0, y_0) and (x_1, y_1) is defined as its Euclidean distance:

$$\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}.$$

Problem A
Age of Elephants
Time limit: 5 seconds

Age of Empires is a classic real-time strategy game from the '90s. A popular game mode involves two players each picking a civilisation to play as, then going head to head in a battle.

There are special units for each civilisation. However, one special unit—the elephant—always provides a decisive victory, unless, critically, the opposing civilisation also has elephants. If neither team has elephants, then neither of them can have a decisive victory against the other. Max wants to pick the best civilisation to play, so he has given you a list of available civilisations and whether they have elephants or not.

For each civilisation, find the number of civilisations they have decisive victories against.

Input

The first line of the input contains one integer N ($1 \leq N \leq 200\,000$), which is the number of civilisations.

The next N lines describe the civilisations. Each of these lines contains two strings s, which is the name of this civilisation, and t ($t \in \{\texttt{yes}, \texttt{no}\}$), which is whether this civilisation has elephants or not. The names of the civilisations contain between 1 and 10 lowercase letters, inclusive. The names of the civilisations are distinct.

Output

For each civilisation in the order given in the input, display the name of the civilisation and how many civilisations they have decisive victories against.

Sample Input 1	Sample Output 1
4 aztecs yes egyptians yes cumans yes japanese yes	aztecs 0 egyptians 0 cumans 0 japanese 0

Sample Input 2	Sample Output 2
4 aztecs yes egyptians no cumans yes japanese no	aztecs 2 egyptians 0 cumans 2 japanese 0

This page is intentionally left (almost) blank.

Problem B
Baby and Choo Choo
Time limit: 2 seconds

Tyson is a baby who loves playing with toy trains. He calls them "choo choo trains" and play with them for hours.

He has a straight train track and trains of various lengths. He wants to fit as many trains on the track as possible. A set of trains fit on the track if the sum of their lengths is at most the length of the track. Unfortunately, because Tyson is a baby, he needs your help to write a program to calculate the maximum number of trains that can fit on the track given its length.[1]

Input

The first line of input contains two integers, L ($1 \leq L \leq 10^9$), which is the length of the track, and N ($1 \leq N \leq 200\,000$), which is the number of trains Tyson has.

The next line describes the trains. This line contains N integers, each is the length of a train and is the inclusive range from 1 to 10^9.

Output

Display the maximum number of trains that can fit on the track.

Sample Input 1	Sample Output 1
10 4 1 9 2 9	2

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

[1]Strangely, baby Tyson is smart enough to use the computer program written in C++, Java, or Python you are about to make him, even if it is written in a programming contest and contains no comments.

This page is intentionally left (almost) blank.

Problem C
Continuous Competitiveness
Time limit: 2 seconds

3/28	1/101	1/47	1/93	6	1/241	3/128	2/29	1/196	1/38	2/149	1/5
2/8	9/298	1/43	1/33	0	1/201	1/257	2/63	0	1/28	1/168	1/11
2/7	1/168	1/32	1/54	0	2/295	1/276	6/152	0	1/21	1/225	1/11
3/21	2/160	1/74	1/53	0	1/177	0	1/107	3	2/65	1/204	1/7
1/27	1/168	1/43	1/82	0	1/256	0	3/166	0	1/18	2/205	1/17
1/10	1/77	1/54	1/125	0	0	2	1/109	0	1/47	2	1/15

Only one team can win the South Pacific Divisionals (this contest), but some other teams might be close enough to the top to be considered competitive. A team is considered competitive if they finish in the top C teams (that is, they have at most $C-1$ teams before them on the scoreboard). Although team members are limited in how many times they can enter over the years, universities are not so limited. A university that has at least one team considered competitive in multiple consecutive years can be considered to be *continuously competitive* over those years.

For example, if a university had some team(s) each year be considered competitive for 5 consecutive years, then that university would be continuously competitive for 5 years. However, if they had some team be competitive for 3 years consecutively, then in the following year no team considered competitive, then in the following two years they again had a competitive team, they would only have been continuously competitive for a maximum span of 3 years.

Given team rankings for consecutive years of the South Pacific Divisionals, your task is to determine the length of the longest span of years for which any university is continuously competitive and the number of distinct universities that have achieved a run that long. Note that two universities that achieve the longest span of years of being continuously competitive may do so in different consecutive spans of years.

Input

The input starts with a line containing two integers N ($1 \leq N \leq 100$), which is the number of consecutive years of team rankings, and C ($1 \leq C \leq 100$), which is the limit of competitiveness, such that a team that is ranked in the top C is considered competitive.

The next N lines describe the standings from the N years in order. Each of these lines contains a single string R ($1 \leq |R| \leq 100$), which is the team rankings for this year with the first place first and the last place last. Each university has been allocated an uppercase or lowercase English letter and each team is identified only by its university name. Note that letter case matters, e.g., `A` and `a` represent distinct universities.

Output

Display the length of the longest run of continuous competitiveness and the number of universities that have achieved at least one run of that length.

Sample Input 1

```
4 2
ABCD
abca
abcb
abcc
```

Sample Output 1

```
3 2
```

Sample Input 2

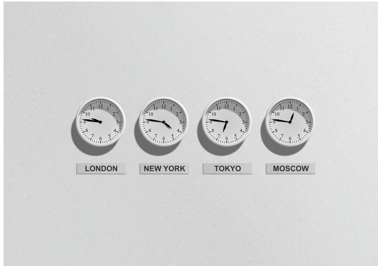
```
4 6
South
Pacific
Programming
Contest
```

Sample Output 2

```
2 3
```

This page is intentionally left (almost) blank.

Problem D
Dream or Reality?
Time limit: 2 seconds



Alice has awoken in a strange land and does not know if it is a dream or reality. In this land, there are many clocks and a stone tablet. The stone tablet has several *clock relations* between clocks of the form X, Y, D, indicating that clock X is exactly D seconds later than clock Y. Note that there may be multiple (not necessarily consistent) clock relations between a pair of clocks.

There is no notion of 'days' for Alice. So if clock X is $24 \cdot 60 \cdot 60$ seconds ahead of clock Y, then they are different times.

If there is a valid assignment of times to each clock such that each clock relation is valid, then Alice is sure that she is awake. Otherwise, she must be dreaming. Given the clock relations, help Alice determine if she is dreaming.

Input

The first line of the input contains two integers N ($1 \leq N \leq 100$), which is the number of clocks, and M ($1 \leq M \leq 10\,000$), which is the number of clock relations. The clocks are numbered 1 to N.

The next M lines describe the clock relations. Each of these lines contains three integers X ($1 \leq X \leq N$), Y ($1 \leq Y \leq N$), and D ($0 \leq D \leq 1\,000$), which means that clock X runs D seconds later than clock Y.

Output

Display `we live in a society` if there is an assignment of times to the clocks that are consistent. Otherwise, display `wake up`.

Sample Input 1	Sample Output 1
3 3 1 2 3 2 3 4 1 3 7	we live in a society

Sample Input 2	Sample Output 2
3 3 1 2 3 2 3 4 1 3 8	wake up

Sample Input 3	Sample Output 3
4 6 1 2 311 1 3 309 4 1 309 3 2 2 4 2 620 4 3 618	we live in a society

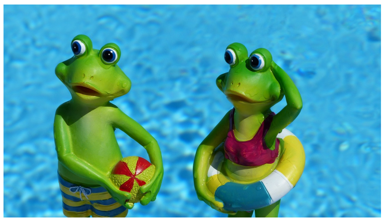
Sample Input 4

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

Sample Output 4

```
we live in a society
```

Problem E
Emailing for Rare Frogs
Time limit: 2 seconds



Eliot and their friends love to collect pictures of rare frogs. There are only a limited number of types of rare frog pictures, and Eliot would like to collect them all.

Each of Eliot's friends has their own collection of rare frog pictures, and may or may not have a picture of each type. It is possible their collection contains no pictures.

Eliot currently has no rare frog pictures but decides to email their friends to ask them to send copies of all their rare frog pictures. Eliot wants to bother the fewest number of their friends but still needs to collect at least one of each rare frog picture type. What is the minimum number of people Eliot has to email in order to get at least one of each type?

Input

The first line of the input contains two integers N $(1 \leq N \leq 16)$, which is the number of friends Eliot has, and M $(1 \leq M \leq 16)$, which is the number of different types of rare frog pictures there are. The types are numbered from 1 to M.

The next N lines describe which rare frog pictures Eliot's friends have. Each of these lines starts with an integer k $(0 \leq k \leq M)$, which is the number of pictures this friend has. The line contains k more integers $f_1, f_2, \dots, f_k$ $(1 \leq f_1 < f_2 < \cdots < f_k \leq M)$, which are the frog types this friend has pictures of.

Output

Display the minimum number of people Eliot has to email in order to get at least one of each type. If it is impossible to get all rare frog pictures, display -1 instead.

Sample Input 1	Sample Output 1
4 3 1 3 2 1 3 1 2 2 1 2	2

Sample Input 2	Sample Output 2
4 3 1 3 1 3 1 2 1 1	3

Sample Input 3	Sample Output 3
3 4 2 1 4 1 2 3 1 2 4	-1

This page is intentionally left (almost) blank.

Problem F
Farmer Gets Rect
Time limit: 2 seconds

Fred and Frances have done it this time! They have been warned several times to not trespass on Farmer Frank's land. Fred and Frances can see Farmer Frank coming towards them with a lantern and they need to hide. They are currently inside the pig's pen, which is a rectangular area. They are able to quickly run to any place within the pig's pen but are not able to leave it.

Farmer Frank is holding a lantern which allows him to see anything that is at most R metres away from him in any direction. He cannot see anything that is more than R metres from him. Farmer Frank will walk in a straight line from one point to another (different) point.

Help Fred and Frances find a place to hide inside the pig's pen so Farmer Frank will not see them at any point on his walk. Once Fred and Frances are in their hiding spot, they may not move.

Input

The first line contains four integers describing the pig's pen. The first two integers, x ($|x| \leq 500$) and y ($|y| \leq 500$), are the coordinates of the bottom-left corner of the rectangle. The last two, W ($1 \leq W \leq 1\,000$) and H ($1 \leq H \leq 1000$), are the width and height of the rectangle, respectively. The pig's pen is the rectangle with corners $(x, y), (x + W, y), (x + W, y + H), (x, y + H)$.

The second line contains five integers describing Farmer Frank's movements. The first integer, R ($1 \leq R \leq 1\,000$), is the distance that the lantern is able to light. The next two integers, x_s ($|x_s| \leq 500$) and y_s ($|y_s| \leq 500$), are the coordinates of Farmer Frank's starting location. The final two integers, x_e ($|x_e| \leq 500$) and y_e ($|y_e| \leq 500$), are the coordinates of Farmer Frank's ending location. $(x_s, y_s) \neq (x_e, y_e)$. Farmer Frank will walk in a straight line between (x_s, y_s) and (x_e, y_e).

Output

Display `Safe` if there is a place for Fred and Frances to hide. Otherwise, display `Oh no!` instead.

Sample Input 1	Sample Output 1
`0 0 3 2` `3 -1 1 1 1`	`Oh no!`

Sample Input 2	Sample Output 2
`1 0 3 2` `3 -1 1 1 1`	`Safe`

This page is intentionally left (almost) blank.

Problem G
Gungle
Time limit: 2 seconds

Letters and Numbers is a game show about letters and sometimes also numbers. *Gungle* is a game you can play with letters. Sadly, it is not played on Letters and Numbers, but it would be nice if it were.

To play Gungle, you must construct a word consisting of only uppercase letters. If you are able to make a word in which each vowel comes immediately after either a ‘G’ or an ‘L’, then you win. The vowels are A, E, I, O, and U.

For example, ‘GUNGLE’ would win, because the ‘U’ comes after a ‘G’ and the ‘E’ comes after the ‘L’. However, ‘GOOGLE’ would not win, since the second ‘O’ comes immediately after the first ‘O’.

Given a word, determine if you would win at Gungle with that word.

Input

The input consists of a single line containing a single string, s, which is the word in question. s contains between 1 and 50 uppercase letters, inclusive.

Output

Display `yes` if s would win at Gungle and `no` otherwise.

Sample Input 1	Sample Output 1
GUNGLE	yes

Sample Input 2	Sample Output 2
ICPC	no

Sample Input 3	Sample Output 3
GLOW	yes

Sample Input 4	Sample Output 4
GOOGLE	no

This page is intentionally left (almost) blank.

Problem H
Helping Picnickers
Time limit: 2 seconds

Eric and Adam are trying to find a good place for a picnic at the park. That was the plan, anyway. A major issue is that it is summer in the southern hemisphere, which means everything is covered with ants.

Luckily, Eric is very good at spotting ants. They have divided the park into an $R \times C$ grid, with each cell being 1 meter by 1 meter. Eric has marked each cell with either an '`A`' to indicate ants or a '`.`' to indicate a clear square with no ants.

The picnic rug that Eric and Adam brought is N meters by M meters. For their picnic, it must be placed axis-aligned on the picnic grid. However, it can be rotated at any 90-degree increment.

For example, if the park has the following grid:

```
A.....
....A.
....A.
......
```

and the rug is 2×3, then all of the following are valid placements ('`X`' denotes the rug):

```
A.....   A.....   A.....
.XX.A.   ..XXA.   ....A.
.XX.A.   ..XXA.   .XXXA.
.XX...   ..XX..   .XXX..
```

Two placements are considered different if they cover a different set of grid squares. How many ways can the rug be placed wholly in the grid without overlapping any ants?

Input

The first line of input contains four integers R ($1 \le R \le 50$), C ($1 \le C \le 50$), N ($1 \le N \le 50$), and M ($1 \le M \le 50$), which are the dimensions of the park and of the rug, respectively.

The next R lines describe the grid. Each of these lines contains a string of length C containing only the characters '.' and '`A`'. A '.' represents a cell that is free of ants and an '`A`' represents a cell with ants.

Output

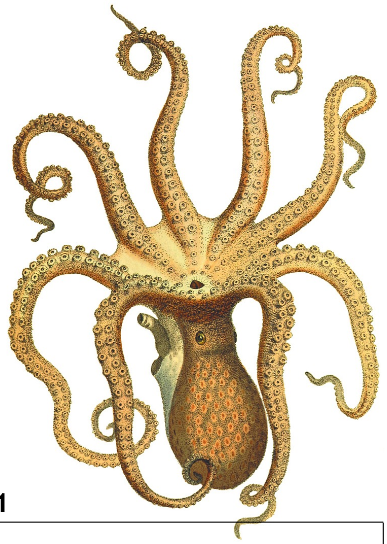
Display the number of different ways a rug can be placed.

Sample Input 1	Sample Output 1
4 6 3 2 A..... ....A. ....A.	10

Sample Input 2	Sample Output 2
1 2 3 4 ..	0

This page is intentionally left (almost) blank.

Problem I
Inky Squbes
Time limit: 2 seconds



Selina the squid loves all numbers, but her favourites are squbes.

An integer n is a square if there is an integer k such that $n = k^2$. Similarly, an integer n is a cube if there is an integer k such that $n = k^3$. An integer n is a sqube if it is both a square and a cube.

Selina would like to know how many squares, cubes, and squbes there are between L and R, inclusive.

Input

The input consists of a single line containing two integers L and R ($1 \leq L \leq R \leq 10^9$), which is the range of numbers Selina is interested in.

Output

Display the number of squares, cubes, and squbes between L and R, inclusive.

Sample Input 1	Sample Output 1
`1 10`	`3 2 1`

Sample Input 2	Sample Output 2
`1 100`	`10 4 2`

Sample Input 3	Sample Output 3
`25 1000`	`27 8 2`

This page is intentionally left (almost) blank.

Problem J
Juicy Posts
Time limit: 5 seconds

Jeremy is trying to write a gossip column in his blog. Gossip is best when it is juicy. Jeremy has blogging software that is able to automatically check if a post is juicy. However, it seems to use some strange rules to judge whether or not a post is juicy. A blog post is juicy if it contains any substring that is on the buzz word list.

For example, if the buzz word list is: 'south', 'pacific', 'programming', and 'contest', then the blog posts: 'jeremylovesfish' and 'ssoouutthh' are not juicy because neither contains any buzz words as substrings. However, the blog post 'jeremeylovesprogrammingonthebeach' is juicy because it contains the substring 'programming'.

Lately, all of Jeremy's stories have been juicy, so his manager is forcing him to write a non-juicy story that contains exactly K characters. How many non-juicy posts of length K are there? Note that a post must be a sequence of lowercase letters.

Input

The first line of input contains two integers N ($1 \leq N \leq 100$), which is the number of buzz words, and K ($1 \leq K \leq 10^{18}$), which is the desired length of the blog post.

The next N lines of input contain the buzz words. Each buzz word contains only lowercase letters. The sum of the lengths of the buzz words is in the inclusive range from 1 to 100.

Output

Display the number of non-juicy posts of length K there are. Because this number can be large, display it modulo 1 000 000 007.

Sample Input 1

```
3 3
abc
def
ghi
```

Sample Output 1

```
17573
```

Sample Input 2

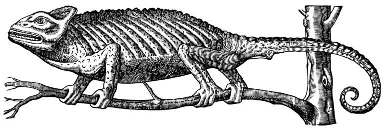
```
4 10
south
pacific
icpc
contest
```

Sample Output 2

```
860830198
```

This page is intentionally left (almost) blank.

Problem K
Kameleon Maze
Time limit: 5 seconds



Pascal the Chameleon is exploring the corn maze where his iguana friends Iggy and Ignia like to frequent on dates. The corn maze is a square grid where some of the cells are blocked off with impassable corn plants and others are cleared out. Pascal can only travel through cells that are cleared out. He can move to a cell in any of the four cardinal directions (up, down, right, and left).

Some of the cleared-out cells contain candies for Pascal to enjoy. There are 9 pieces of candy each labeled using the integers $1, 2, \ldots, 9$. Pascal is very particular about the order he eats his candy, so he must eat the candies in increasing label order.

Every second, Pascal can either move to an adjacent cell in the grid or eat a single piece of candy, but not both. As Pascal is a chameleon, he has a long tongue he can use to grab candy. Pascal can grab and eat a piece of candy if it is at most k cells away straight in one of the four cardinal directions and there is no blocked-off cell between him and the candy. Pascal can also eat the candy if he is in the same cell as it. His tongue is able to pass through cells containing candies that should be eaten later without eating it.

How long will it take for Pascal to eat all of the candies in order if Pascal traverses the maze optimally?

Input

The first line of input contains three integers r, c ($10 \leq r \cdot c \leq 200\,000$), which are the number of rows and columns of the rectangular grid representing the maze, and k ($0 \leq k \leq 200\,000$), which is the maximum distance Pascal can reach with his tongue.

The next r lines describe the grid. Each of these lines contains a string of length c containing only the characters '`.`', '`#`', '`S`', and the digits '`1`', '`2`', ..., '`9`'. A '`.`' represents a cell that is cleared out, a '`#`' represents a cell blocked off with corn plants, an '`S`' represents a cleared out cell where Pascal starts in the maze, and each number represents the a cleared out cell containing a candy Pascal must collect (in their numerical order).

The character '`S`' and the digits '`1`', '`2`', ..., '`9`' all appear in the grid exactly once. It is guaranteed that Pascal can reach each of the 9 candies in the maze.

Output

Display the minimum amount of time in seconds Pascal can take to eat all of the candy.

Sample Input 1
```
2 5 1
S1234
56789
```
Sample Output 1
```
17
```

Sample Input 2
```
4 5 2
98.67
.S.1.
.#4#.
32..5
```
Sample Output 2
```
15
```

Sample Input 3
```
1 10 10
S987654321
```
Sample Output 3
```
9
```

This page is intentionally left (almost) blank.

Problem L
Lazy Students
Time limit: 3 seconds

Larry and Lena used to study at a very relaxed pace, taking only one course each year. They ended up taking the exact same set of courses — the easiest ones needed to get their degrees — albeit in (possibly) different orders.

Some courses have prerequisites, meaning that a certain course must be taken before you can take this course. Given the order in which Larry and Lena took their courses (both satisfying all prerequisite requirements), determine the maximum number of prerequisites that could exist between courses.

Input

The first line of the input starts with an integer n ($2 \leq n \leq 200\,000$), the number of courses. Courses are numbered 1 to n, with 1 being the first course taken by Larry and n the last.

The second line contains a permutation of the integers from 1 to n, inclusive, describing the order in which Lena took these courses.

Output

Display the maximum number of prerequisites.

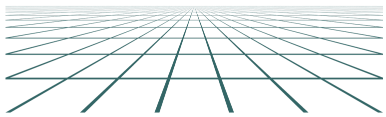
Sample Input 1	Sample Output 1
3 1 2 3	3

Sample Input 2	Sample Output 2
3 3 2 1	0

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

This page is intentionally left (almost) blank.

Problem M
Master of Jenga
Time limit: 2 seconds



Jessica is a 2D-Jenga master! The game of 2D-Jenga is played on a rectangular grid and is played by placing bricks that are of width k and height 1 onto the grid. Each row must have exactly one brick on it and that brick must cover exactly k consecutive columns of that row. Each brick, except the one on the bottom row, must balance on the brick in the row directly underneath it. Jessica is such a skilled player that she can make a brick balance on another as long as the two bricks share at least one column.

Jessica is so good at the game that no one will play with her anymore, so she has come up with a personal challenge. She has given each grid cell a positive integer value. The number of points she scores is the total of all cells that she covers. Her goal is to place the bricks in such a way that maximizes the number of points.

Given the grid of values, give one possible configuration that maximizes the number of points.

Input

The input starts with a line containing three integers R ($2 \le R \le 50$), which is the number of rows in the grid, C ($2 \le C \le 50$), which is the number of columns in the grid, and k ($2 \le k \le C$), which is the length of each block.

The next R lines describe the grid. Each of these lines contains C positive integers representing the value of the grid cells. Each cell's value is a positive number that is at most 10^9.

Output

Display a valid configuration of the bricks that maximizes the score. An `X` represents a brick and a `.` represents a cell without a brick. If there are multiple possible solutions, any will be accepted.

Sample Input 1

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

Sample Output 1

```
..XX
.XX.
..XX
```

Sample Input 2

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

Sample Output 2

```
..XX
.XX.
XX..
```

Sample Input 3

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

Sample Output 3

```
XX.
.XX
XX.
XX.
XX.
.XX
XX.
.XX
```

Sample Input 4	Sample Output 4
2 4 2 9 9 1 1 1 1 9 9	XX.. .XX.