Buddhist Sin Tak College Competitive Programming Team

Mid-Term Examination (Junior Group)

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
A	Score Calculation II	1.000 s	256 MB	10 checkpoints
B	Decode II	1.000 s	256 MB	7 + 3 + 8 + 8 + 74
C	Dynamic Walking	1.000 s	256 MB	10 checkpoints
D	Run Lads Run	1.000 s	256 MB	28 + 72

Notice:

Unless otherwise specified, inputs and outputs shall follow the format below:

- One space between a number and another number or character in the same line.
- No space between characters in the same line.
- Each string shall be placed in its own separate line.
- Outputs will be automatically fixed as follows: Trailing spaces in each line will be removed and an end-of-line character will be added to the end of the output if not present. All other format errors will not be fixed.

Only **C++** is allowed to be used in this examination.

C++ programmers should be aware that using C++ streams (`cin` / `cout`) may lead to I/O bottlenecks and substantially lower performance.

For some problems 64-bit integers may be required. In C++ it is `long long` and its token for `scanf` or `prinf` is `%lld`.

All tasks are divided into subtasks. You need to pass all test cases in a subtask to get points.

A

Score Calculation II

Time Limit: 1.000 s Memory Limit: 256 MB

You, *pepper1208* and *rina__owo* formed a team to participate in ICPC (International Collegiate Programming Contest). There were N teams in total participated in this round of contest, numbered from 1 to N. Also, there were M problems in the contest. On the t-th minute ($1 \le t \le T$), the team i submitted their solution to problem j, which the solution was correct (verdict: `AC`) or incorrect (verdict: `WA`).

The total score of a team is the sum of the fastest solution time in minute plus a penalty of 20 minutes per incorrect attempt for all solved problems. If a problem is not solved (which the team does not submit any correct solution for the problem), the attempts do not give any penalty to the score.

After calculating the total score of all teams, the ranking of the contest can be obtained as follows:

1. The team with higher number of solved problems ranked higher.
2. If both team have the same number of solved problems, the team with less total score ranked higher.
3. If both team have the same number of solved problems and the same total score, the team with the earlier submission time for the latest correct submission ranked higher.
4. If both team have the same number of solved problems, same total score, and same submission time for the latest correct submission, the team with lower team number ranked higher.

Given that there are Q submissions made by all teams during the contest. Now, *pepper1208* and *rina__owo* asks you for the ranking of our team. Could you answer them?

INPUT

Input $Q+1$ lines.

The first line contains five integers N, M, T, Q and x. x is the team number of your team.

Each of the following Q lines contains four integers t, i, j and v, which in t-th minute, team i submitted their solution to problem j with the verdict v (either `AC` or `WA`).

It is guaranteed that the Q lines are input in the ascending order of the submission time.

It is guaranteed that there will be no more than one submission for a team in one minute.

OUTPUT

Output two integers in a single line separated by a whitespace, which is the total score in minute and the ranking of your team respectively.

SAMPLES

	Input	Output

1	`3 3 5 5 1`
	`1 1 2 AC`
	`2 2 1 WA`
	`3 3 3 AC`
	`4 1 3 WA`
	`5 1 3 AC`
		`26 1`

When t = 1, team 1 submitted their solution to problem 2, with `AC` verdict.
When t = 2, team 2 submitted their solution to problem 1, with `WA` verdict.
When t = 3, team 3 submitted their solution to problem 3, with `AC` verdict.
When t = 4, team 1 submitted their solution to problem 3, with `WA` verdict.
When t = 5, team 1 submitted their solution to problem 3, with `AC` verdict.

The total score for team 1 is $1+5+20=26$, with 2 problems solved.
The total score for team 2 is 0, with 0 problems solved,
The total score for team 3 is 3, with 1 problem solved.

SUBTASKS

For all test cases,

$1 \le N \le 10$

$1 \le M \le 10$

$1 \le T \le 200$

$1 \le Q \le 2000$

Points are given per **checkpoint** in this problem. You can get the point of the checkpoint when you pass them.

There are 10 checkpoints in total, each carry 10 points.

B Decode II

Time Limit: 1.000 s Memory Limit: 256 MB

Dr. Jones is a great mathematician and a well-known archaeologist. One day, Dr. Jones and you, his assistant, found an ancient treasure box with a mysterious password lock, which requires you to enter an integer as the password (denoted as x).

After investigation, Dr. Jones found that the password will be correct only if $x + \mathrm{rev}(x)$ is a **palindrome**. Here, the function $\mathrm{rev}(x)$ denotes the integer which is the reverse order of the integer x. For example, $\mathrm{rev}(123) = 321$ and $\mathrm{rev}(24212580) = 8521242$. If x is a palindrome, then $x = \mathrm{rev}(x)$.

Dr. Jones will give you Q integers. For each integer, he wants you to determine whether it is a correct password for the lock. If you do not get all correct, you will be fired. Try your best and not be fired!

INPUT

Input $Q+1$ lines.

The first line contains an integer Q.

Each of the following Q lines contains an integer x.

OUTPUT

Output Q lines.

For each x, output `Yes` if x is a correct password. Otherwise, output `No`.

SAMPLES

	Input	Output
1	2 123 555	Yes No

When x = 123, x + rev(x) = 444, which is a palindrome.

When x = 555, x + rev(x) = 1110, which is not a palindrome.

SUBTASKS

For all test cases,

$1 \le Q \le 100$

$0 \le x \le 10^{5000}$

Points are given per **subtask** in this problem. You have to pass all the checkpoint in the subtask in order to get the points of the subtask.

	Score	Constraints
1	7	$x \le 100$
2	3	$x \le 10^{14}$
3	8	Every digit in $x < 5$
4	8	Every digit in $x > 5$
5	74	No additional constraints

C Dynamic Walking

Time Limit: 1.000 s Memory Limit: 256 MB

There is a longlasting classic puzzle: given a $N \times M$ grid, you start from the top left corner (1, 1) and you can only move rightwards or downwards 1 unit each time. Some obstacles may exists in the grid such that you cannot pass through them. How many ways are there for you to travel from the top left corner to the bottom right corner (N, M) of the grid?

Some proposes that *dynamic programming* helps us solve this puzzle with the trick of memoization (not memorization!). Let $dp[i][j]$ stores the number of paths from (1, 1) to (i, j). Obviously, we can obtain the relation:

$$dp[i][j] = dp[i-1][j] + dp[i][j-1]$$

Trivially, $dp[1][1] = 1$ as you have only one way to travel from (1, 1) to (1, 1). However, the relation above is incorrect with the consideration of obstacles inside the grid. Therefore, it is time for you to fully solve the *Dynamic Walking*!

INPUT

The first line contains two integers, N and M.

Each of the following N lines contains M characters, to show the full grid.

- If the character is `.` , it indicates an empty space that you can pass through.
- If the character is `#` , it indicates an obstacle that you cannot pass through.

It is guaranteed that there is no obstacle in the top left corner and the bottom right corner in the grid.

OUTPUT

Output one integer, which is the total number of possible paths for you to travel from (1, 1) to (N, M).

SAMPLES

	Input	Output

1	3 3
	...
	...
	...
		6

2	3 4 ..#. .... #...	6

SUBTASKS

For all test cases,

$1 \le N, M \le 25$

Points are given per **checkpoint** in this problem. You can get the point of the checkpoint when you pass them.

There are 10 checkpoints in total, each carry 10 points.

D

Run Lads Run

Time Limit: 1.000 s Memory Limit: 256 MB

Bob is rushing through N blocks on the straight lines. Each blocks can be classified as two type: type A and type B. He runs through all blocks, but he thinks it is too slow. After hearing the song "Run Lads Run", he receives a superpower! When he runs through two blocks with the same type, he receives a single time of speed boost!

Nevertheless, there are M blocks that remain unclassified. You, as a great block-master, can modify any blocks as type A or type B. After modifying all unclassified blocks, Bob wants to know all possible number of speed boost he will receive to help him run better. Now, it is your job to answer him the *speedy* fact.

INPUT

Input two lines.

The first line contains an integer N.

The second line contains a string S. The string contains of characters `A`, `B` and `X` only, which is the type of the blocks in ascending order:

- `A`: The type of the block is type A
- `B`: The type of the block is type B
- `X`: The block is unclassified as type A or type B

OUTPUT

Output $K + 1$ integers.

In the first line output an integer K, which is the number of distinct number of speed boost Bob can receive.

On the next K lines, output the number of speed boost Bob can receive, **in ascending order**.

SAMPLES

	Input	Output
1	4 ABBX	2 1 2

If the blocks are “`ABBA`”, then Bob will receive one speed boost when travelling from the second block to the third block.

If the blocks are “`ABBB`”, then Bob will receive two speed boosts when travelling from the second block to the third block and from the third block to the fourth block.

	Input	Output
2	9 XBAXBAXBA	2 2 3
3	10 AXXXXXBAXB	3 2 4 6

Note that it is not required to classify all unclassified blocks to be the same type.

SUBTASKS

For all test cases,

$1 \le N \le 2 \times 10^5$

Points are given per **subtask** in this problem. You have to pass all the checkpoint in the subtask in order to get the points of the subtask.

	Score	Constraints
1	28	$N \le 10$
2	72	No additional constraints