

Operation Imaginary I

INFORMATION

Date	15 April 2022
Time	13:40 - 17:40 (4 hours)
Full Score	500

PROBLEMS

	Problem	Limit		Score	
		Time	Memory	Subtasks	Total
TW2210	Judge-team-work	1s	256MB	20 + 30	50
TW2211	Stat-mit-base	1s	256MB	6 + 12 + 24 + 23 + 12 + 13 + 12 + 28 + 20	150
TW2212	Front-end-ing	1s	256MB	20 + 20 + 20 + 40	100
TW2213	Yee-launch-must	1s	256MB	35 + 25 + 25 + 15	100
TW2214	Pass-word-bot	1s	256MB	20 + 30 + 20 + 30	100

REMINDERS

- All tasks are divided into subtasks, your program must pass all test cases in a subtask to get points.
- Unless otherwise specified, your output will be automatically fixed as follows:
 - Trailing spaces in each line will be removed.
 - An end-of-line character will be added to the end of the output if not present.
- 64-bit integers may be required in some problems, use `long long` in C++ if needed.

TW2210 Judge-team-work

Score	50
Time Limit	1s
Memory Limit	256MB

Daniel, the TWGSS OI Team leader, had a dream for a long time — build an online judge for OI! But it is not an easy task to do, as there are many aspects that the developer(s) of the judge must be good at. Therefore, he first approached Jack, another form 5 “old person”, to help him finding people and form a team. Jack said, “Wow! Such a difficult job!”, but they still found Duncan to join the team. “Oh no, we know nothing about servers!” the newly formed team said. Therefore, they asked everyone in the OI Team who wanted to “build a website”, and a person — Ryan replied! Daniel and Jack, ran to form 2 classroom and found Ryan, asking if he was willing to join the team, and the reply they got was positive! As a result, a 4-people team, with Daniel, Jack, Duncan and Ryan is formed! They started the long journey of making the judge.

What is the relationship between the story above and this problem? Maybe a little bit. In this problem, you are given the number of members in the team and their skill level in different aspects. For each aspect, the work is done by the members with the highest skill level in that aspect, which determined the “aspect level” of that aspect. The sum of all “aspect level” determined the total “judge level” (i.e. how good is the judge). Note that one person can do works of multiple aspects, and can also have no work entirely.

Can you help the team to calculate what will the “judge level”?

INPUT

The first line consists of two integers N and M , separated by a space, indicating number of members in the team and number of aspects respectively.

For the next N lines, the i^{th} ($1 \leq i \leq N$) line contains M integers, each separated by a space, the j^{th} ($1 \leq j \leq M$) integer S_{ij} indicates the skill level of member i in aspect j .

OUTPUT

Output an integer, the “judge level” of the judge.

SAMPLE TESTS

	Input	Output
1	4 5 25 22 10 11 5 13 15 19 5 24 13 5 8 3 8 7 5 15 23 20	113

The 1st member will do works of aspect 1 and 2;

the 2nd member will do works of aspect 3 and 5;

the 3rd member has no work to do;

the 4th member will do works of aspect 4.

“Judge level” = $25 + 22 + 19 + 23 + 24 = 113$.

CONSTRAINTS

$$1 \leq N, M \leq 1000$$

$$0 \leq S_{ij} \leq 10^9 \text{ for all } i, j$$

SUBTASKS

	Points	Constraints
1	20	$1 \leq N, M \leq 10$ $0 \leq S_{ij} \leq 100$ for all i, j
2	30	No additional constraints

TW2211 Stat-mit-base

Score	150
Time Limit	1s
Memory Limit	256MB

Daniel, the backend developer, was making the judge of the judge (you know, the judge need judge to judge things). Sadly, he had no time to deal with the submission system! Therefore, he asked you to help him with that. You have two choices: Use SQL or build a system yourself. You decided to challenge yourself, so you choose to go with the difficult and complicated way — build a system yourself!

Given N submission data of a problem, the i^{th} ($1 \leq i \leq N$) data represents the submission with ID of i , the input details specified as follows:

- 1 T_i , the submission time.
- 2 V_i , the verdict of the submission, which is a string consists of 2 to 5 uppercase English characters.
- 3 R_i , the runtime of the code (in millisecond).

There are 5 kinds of queries:

- 1 `count  $v$   $t_l$   $t_r$` : given a string v , two integers t_l and t_r . Find and output the number of submissions with verdict v between time t_l and t_r (inclusive).
- 2 `mean  $t_l$   $t_r$` : given two integers t_l and t_r . Find the mean of runtime of all submission with verdict `AC` between time t_l and t_r (inclusive).
 - a If there are no submissions in this time range with verdict `AC`, output `no mean`.
 - b Otherwise, output the rounded down value of the mean found.
- 3 `median  $t_l$   $t_r$` : given two integers t_l and t_r . Find the median of runtime of all submission with verdict `AC` between time t_l and t_r (inclusive).
 - a If there are no submissions in this time range with verdict `AC`, output `no median`.
 - b Otherwise, output the rounded down value of the median found.
- 4 `mode  $t_l$   $t_r$` : given two integers t_l and t_r . Find the mode of runtime of all submission with verdict `AC` between time t_l and t_r (inclusive).
 - a Output `no mode` (brain fog) if no mode (i.e. all runtime appear in this time range with verdict `AC` are mode or there are no submissions in this time range with verdict `AC`)
 - b Otherwise, output the smallest mode.
- 5 `update  $d$   $v$   $r$` : given an integer d , a string v , and an integer r . Update the submission with submission ID d . Update its verdict to v , runtime to r , and the submission time is unchanged. It is guaranteed that the submission with submission ID d exists.

NOTE

Consider a set of data $\{a_1, a_2, \dots, a_k\}$. The statistical data

- mean is defined as the average, (i.e. sum of $\{a_i\}$ divided by the number of data). In this problem, you are required to find the round down value of mean.
- median is defined as the middle term of the sorted sequence of $\{a_i\}$. Formally, suppose $\{s_1, s_2, \dots, s_k\}$ is the sorted sequence of $\{a_i\}$ where $s_1 \leq s_2 \leq \dots \leq s_k$. Then if k is odd, the median is given by $s_{\frac{k+1}{2}}$; if k is even, the median is given by $\frac{1}{2}(s_{\frac{k}{2}} + s_{\frac{k}{2}+1})$. In this problem, you are required to find the round down value of median.
- mode is all the data which appears the most time. In this problem, you are required to find the smallest mode.

Example:

- $\{1, 2, 5, 7, 8\}$
Mean: $\frac{1+2+5+7+8}{5} = 4.6$ (4 when rounded down)
Mode: As all number have the same frequency of occurrence of 1 time, there is no mode.
Median: 5
- $\{1, 2, 3, 3\}$
Mean: $\frac{1+2+3+3}{4} = 2.25$ (2 when rounded down)
Mode: 3
Median: $\frac{2+3}{2} = 2.5$ (2 when rounded down)

INPUT

The first line of input consists of two integers N and Q , the number of submission data and the number of queries.

For the i^{th} line of the next N lines, it contains an integer T_i , a string V_i , and an integer R_i and indicating the submission time, the verdict, and the runtime respectively of the i^{th} submission.

Each of the next Q lines contains a query, which the format is specified above.

OUTPUT

For each query with output specified, output it in a new line.

SAMPLE TESTS

Input	Output
3 7	1
2 AC 3	4
3 TLE 1000	no mode
5 AC 5	3
count AC 1 3	4
/ mean 2 5	
update 2 AC 998	
mode 2 5	
update 2 AC 3	
mode 2 5	
median 3 5	

CONSTRAINTS

$$1 \leq N, Q \leq 50000$$

$$1 \leq R_i, r \leq 1000$$

$$1 \leq T_1 \leq T_2 \leq \dots \leq T_N \leq 10^9$$

$$1 \leq t_l \leq t_r \leq 10^9$$

SUBTASKS

	Points	Constraints
1	6	$1 \leq N, Q, T_i \leq 50$ All queries are <code>count</code>
2	12	$1 \leq N, Q, T_i \leq 50$ All queries are <code>count</code> or <code>update</code>
3	24	$1 \leq N, Q, T_i \leq 50$
4	23	$t_l = 1, t_r = 10^9$ for all queries There is no <code>median</code> query
5	12	$t_l = 1, t_r = 10^9$ for all queries
6	13	All queries are <code>count</code> , <code>mean</code> or <code>update</code>
7	12	There is no <code>update</code> query
8	28	$1 \leq N, Q \leq 5000$
9	20	No additional constraints

TW2212 Front-end-ing

Score	100
Time Limit	1s
Memory Limit	256MB

As we all know, online judge exists as a website. With the “functional” judge is not enough, the Times New Roman font and the lack of design is unacceptable for many people. The frontend developer, Jack, decided to do some work.

When he started to investigate into the html file. He discovered that all styles are hardcoded! He felt very sad, as sorting out the code is very annoying and confusing. As a friend of Jack, you decided to help him. Given the html “code”, can you help him to extract out the visible contents?

Html in this problem works as follows (which is not exactly resembling real html):

- 1 The exact text `<br>` is the end-of-line symbol, only output new-line when `<br>` is present. No newline symbols or spaces should be included in `<br>`.
- 2 Characters inside `<>` are tags, which is not the visible contents of the file. It is guaranteed that there is no other `<`, `>` in tags.
Example: Visible contents of `<span id="word">he llo</span>` should be `he llo`.
- 3 More than one spaces should be grouped into one. Multiple spaces spanning multiple lines should also be merged into one space.

Note that the above procedure run in the order.

Example: Visible contents of `Hello <span id="word"> World!</span>` is `Hello World!` instead of `Hello World!` (i.e. the spaces after `Hello` and `World` are merged).

INPUT

There are unknown lines of input, which is the html “code”.

It is guarantee that the html “code” is valid.

OUTPUT

The visible content of the html “code”.

Autofix disabled for this task. Your raw output will be used for judging. DO NOT add extra `endl` after your output unless it is specified in the input.

SAMPLE TESTS

Input	Output
<pre><div id="hi">Hello, World!</div> :) New Day!
 This is
 fun! HTML fun, right?</pre>	<pre>Hello, World!:) NewDay! This is fun!HTML fun, right?</pre>

CONSTRAINTS

$1 \leq \text{Length of input} \leq 256$

SUBTASKS

	Points	Constraints
1	20	No. of lines in input = 1 The input does not contain spaces and <code>
</code>
2	20	The input does not contain spaces and <code>
</code>
3	20	The input does not contain spaces
4	40	No additional constraints

TW2213 Yee-launch-must

Score	100
Time Limit	1s
Memory Limit	256MB

The judge was about to launch! Duncan, who uses the “Yee” image for his profile picture, felt excited! As the Cantonese pronunciation of “2” is very similar to “Yee”, he wanted the launch date of the judge has as many “2” as possible.

As it will be disastrous if the judge is unfinished when launch, and it will be sad if the judge is finished way before its launch. Therefore the “Yee index” is determined as follows:

- 1 The “Yee index” will be 10 times of the number of 2 in its date (in the format of DD-MM-YYYY).
- 2 The “Yee index” will decreased by 1 for every day it is before OR after the expected finish date of the judge.

Given the expected finish date of the judge, determined the launch date of the judge, which is the date with the highest “Yee index”, if multiple dates have the highest “Yee index”, the earliest one is the launch date.

Note: A year is a leap year if it is divisible by 4 but not divisible by 100. But years that are divisible by 400 are also leap years, although they are divisible by 100.

INPUT

The input consists of only 1 line, which is the expected finish date of the judge.

The date is in the format of DD-MM-YYYY.

OUTPUT

Output 1 line, the launch date of the judge, in the format of DD-MM-YYYY.

SAMPLE TESTS

- | | Input | Output |
|---|------------|------------|
| 1 | 13-02-2022 | 22-02-2022 |
- 22-02-2022 have 6 2s in the date and it is 9 days after the expected finish day, therefore it has a “Yee index” of $6 \times 10 - 9 = 51$.
- | | | |
|---|------------|------------|
| 2 | 07-01-2022 | 02-01-2022 |
|---|------------|------------|
- 02-01-2022, 12-01-2022 and 22-01-2022 all have a “Yee index” of 35, as 02-01-2022 is the earliest, it is the launch date.

CONSTRAINTS

Both the expected finish date and the launch date of the judge are between year 1900 and year 9999 (inclusive).

It is guaranteed that the expected finish date given is valid.

SUBTASKS

	Points	Constraints
1	35	Both the expected finish date and the launch date of the judge are in the year 2022 The launch date is on or after the expected finish date
2	25	Both the expected finish date and the launch date of the judge are in the year 2022
3	25	Both the expected finish date and the launch date of the judge are between the year 2021 and 2023 (inclusive)
4	15	No additional constraints

TW2214 Pass-word-bot

Score	100
Time Limit	1s
Memory Limit	256MB

The judge has a Discord server for communication across the team. Ryan, the long-time Discord user, had the idea of making a bot to get the password of users! (Don't panic, this bot doesn't exist in real-life, all the passwords are hashed. But there is a bot reminding the team when someone changed their password though.)

The bot will perform two types of action:

- 1 Setting / Changing Password (indicate by **S**)
Given a username U_i and a password P_i .
If the user with the username U_i does not exist in the judge before the action, a new user with the username U_i will be registered with the password of P_i .
If the user with the username U_i already exists in the judge before the action, the password of that user will be changed to P_i .
- 2 Requesting Password (indicate by **R**)
Given a username U_i .
If the user with the username U_i already exists in the judge, output the password of that user.
Otherwise, output **User Not Found**.

U_i and P_i are string that does not contain spaces. Note that the bot is case-sensitive. (i.e. Upper-case and lower-case letters are different)

INPUT

The first line of input contains an integer N , indicating the number of actions.

For the i^{th} line of the next N lines of the input, the first character A_i indicates the type of the i^{th} action:

- If A_i is **S**, two strings U_i and P_i follows, separated by a space.
- If A_i is **R**, only one string U_i follows.

OUTPUT

For every action where A_i is **R**, output a new line with the result of the i^{th} action.

SAMPLE TESTS

Input	Output
13	User Not Found
S Daniel Qwerty	HTML_problem_fun?
S Jack HTML_problem_fun?	yee
R Duncan	User Not Found
S Duncan yee	Qwerty
S Ryan Password123	Qwertypi
R Jack	YeeYeeYee
R Duncan	
R rYAN	
S Duncan YeeYeeYee	
S daniel Qwertypi	
R Daniel	
R daniel	
R Duncan	

CONSTRAINTS

$$1 \leq N \leq 10^5$$

$$1 \leq \text{Length of } U_i \text{ and } P_i \leq 24$$

SUBTASKS

	Points	Constraints
1	20	$1 \leq N \leq 100$ For every action which A_i and A_j are $\boxed{S}$, $U_i \neq U_j$ if $i \neq j$
2	30	$1 \leq N \leq 100$
3	20	For every action which A_i and A_j are $\boxed{S}$, $U_i \neq U_j$ if $i \neq j$
4	30	No additional constraints