Important message on plagiarism

The single most important point for you to realize before the beginning of your studies at ShanghaiTech is the meaning of "plagiarism":

> Plagiarism is the practice of taking someone else's work or ideas and passing them off as one's own. It is the misrepresentation of the work of another as your own. It is academic theft; a serious infraction of a University honor code, and the latter is your responsibility to uphold. Instances of plagiarism or any other cheating will be reported to the university leadership, and will have serious consequences. Avoiding any form of plagiarism is in your own interest. If you plagiarize and it is unveiled at a later stage only, it will not only reflect badly on the university, but also on your image/career opportunities.

Plagiarism is academic misconduct, and we take it very serious at ShanghaiTech. In the past we have had lots of problems related to plagiarism especially with newly arriving students, so it is important to get this right upfront:

You may ...

- ... discuss with your peers about course material.
- ... discuss generally about the programming language, some features, or abstract lines of code. As long as it is not directly related to any homework, but formulated in a general, abstract way, such discussion is acceptable.
- ... share test cases with each other.
- ... help each other with setting up the development environment, etc.

You may not ...

- ... read, possess, copy or submit the solution code of anyone else (including people outside this course or university)!
- ... receive direct help from someone else (i.e. a direct communication of some lines of code, no matter if it is visual, verbal, or written)!
- ... give direct help to someone else. Helping one of your peers by letting him read your code or communicating even just part of the solution in written or in verbal form will have equal consequences.
- ... gain access to another one's account, no matter if with or without permission.
- ... give your account access to another student. It is your responsibility to keep your account safe, always log out, and choose a safe password. Do not just share access to your computer with other students without prior lock-out and disabling of automatic login functionality. Do not just leave your computer on without a lock even if it is just for the sake of a 5-minute break.
- ... work in teams. You may meet to discuss generally about the material, but any work on the homework is to be done individually and in privacy. Remember, you may not allow anyone to even just read your source code.

With the Internet, "paste", and "share" are easy operations. Don't think that it is easy to hide and that we will not find you. We have just as easy to use, fully automatic and intelligent tools that will identify any potential cases of plagiarism. And do not think that being the original author will make any difference. Sharing an original solution with others is just as unethical as using someone else's work.

CS100 Homework 3 (Spring 2023)

Deadline: 2023-03-15 (Wed) 23:59:59

Late submission will open for 24 hours after the deadline, with 50% point deduction.

If you get full marks in this assignment by no more than 40 submission attempts and **also solve the challenge in problem 4**, you can earn special OJ displays and a "1-case protection" that can be used in further assignments to cancel one testcase failure. See Piazza or OJ dashboard for more information.

Problem 1: Chihiro Fujisaki and Caesar Cipher

"If I don't do something, nothing's ever going to change."

Description:

Chihiro Fujisaki, the Ultimate Programmer, is developing a new kind of artificial intelligence called Alter Ego. Now he is training the artificial intelligence secretly in order to prevent Monokuma spotting it.

Fujisaki is training Alter Ego with Caesar cipher, an ancient encryption algorithm. With a fixed number as `key`, for example, 3, Caesar cipher replaces each letter in the plaintext a letter 3 positions down the alphabet. In this case, the message `box` becomes `era`, because going down 3 letters turns `b` into `e` and turns `x`, going through the alphabet, into `a`.

Fujisaki's messages contain uppercase letters, lowercase letters, digits, and other characters like spaces and punctuations. He doesn't want to scramble his messages, so he keeps the cases and encrypts uppercase letters, lowercase letters and digits respectively (so that `Z` encodes to `C` and `9` to `2`). All other characters, including spaces and punctuations, are not encoded.

Monokuma has discovered this secret, and has obtained the secret `key`. Please help Monokuma to decrypt Fujisaki's messages.

Input Format:

The input contains two lines:

The first line contains a number n, the `key` for Caesar cipher. $n > 0$ shifts the message right (going down the alphabet), and $n < 0$ shifts it left (going up the alphabet).

The second line contains a non-empty string `ciphertext`, the encoded text, ending with a line break character (`'\n'`). Only uppercase letters, lowercase letters, and digits have been encoded.

It is guaranteed that $|n| \leq 10^6, |ciphertext| \leq 10^6$.

Output Format:

You need to print the decoded message in one line.

Samples

Sample 1

Input

```
8
Qv KA988, xtioqizqau qa bismv dmzg amzqwca! Mdmzg gmiz, uivg abclmvba omb kicopb
ivl omb xcvqapml!
```

Sample 2

Input

```
-30
Zk JKP odwna ukqn ykza sepd kpdano, zk JKP yklu ykza bnki Ejpanjap kn kxpwej ykza
bnki opqzajpo ej lwop uawno, wjz zk JKP qoa YdwpCLP ej dkiaskngo! Sa ydayg whh
ukqn oqxieooekjo wjz sa SEHH gjks!
```

Sample Outputs...?

Whoops, ciphers are no more secret if we tell you the answers! Why not figure out yourself? Hints: the above samples all decode to proper English sentences.

Problem 2: Chihiro Fujisaki and Keyworded Caesar Cipher

"I'm scared, but... I can handle it. I... don't really understand why, but...

When I think about everyone else, my courage starts to grow... !"

Description

Monokuma has cracked Chihiro Fujisaki's Caesar Cipher. The Ultimate Programmer Fujisaki is working on a more secure code called **"Keyworded Caesar Cipher"**. Let Fujisaki explain his genius ideas to you:

"My keyworded Caesar cipher no longer has a numerical `key`, but a `keyword`, for example, `"wednesday"`. Just as the original Caesar cipher creates a shifted alphabet for encoding letters, so too does my keyworded Caesar cipher have one, and this encoding alphabet is created by the `keyword` `"wednesday"`.

"I first turn all letters into lowercase, and remove duplicate letters in the keyword, keeping only their first occurences. Now `"wednesday"` has become `"wednsay"`.

"Then I write the alphabet `a` through `z` in one line, and my new keyword `"wednsay"` beneath it.

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
w e d n s a y
```

"And finally, I fill the rest of my encoding alphabet starting from the last letter `y`. However, **letters in the keyword are already used**. I should skip these letters when filling the rest, because I cannot let two letters encode to a same value.

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
w e d n s a y z b c f g h i j k l m o p q r t u v x
```

"Alright. I can use it to encode `a` to `w`, `z` to `x`, and my name `Chihiro` to `Dzbzbmj`! For uppercase letters I can do the same, but not for digits any more. It's no big deal. I know in homework 1 you turned numbers into English words, right? (laughs)"

How nice would it be, if only you and he know the secret! Monokuma has heared all your conversation and learned his new method. More importantly, Monokuma has obtained the `keyword`! Please help Monokuma again to decrypt Fujisaki's messages.

Input Format

The input contains two lines:

The first line contains a non-empty string `keyword`, containing only uppercase and lowercase letters.

The second line contains a non-empty string `ciphertext`, the encoded text, ending with a line break character (`'\n'`). Only uppercase and lowercase letters have been encoded.

It is guaranteed that $1 \leq |keyword| \leq 100, |ciphertext| \leq 10^6$.

Output Format

You need to print the decoded message in one line.

Samples

Sample 1

Input

```
CPPistheBESTlanguage
(1) Ewws mcq wh btgxlve: sldijdd qwjz lstcd mlfb qwjz hzltvsd. (2) Pcs mcq wh
btgxlve: elkt qwjz iwst fw dwutwvt.
```

Sample 2

Input

```
ProgrammingIsFun
Qyhc oyga pxg qyhc pooyhxed pca qyhc daocae. Qyh npja ena cadzyxdsrsvseq ey
zcyeaoe enaw. Gy XYE vae pxqyxa pooadd qyhc pooyhxe, hda qyhc oywzheac, yc vyyu
pe qyhc oyga!
```

Sample Outputs...?

Secret, of course! Again, they all decode to proper English sentences.

Hints

You may find some functions in `<string.h>` useful, for example, `strlen`, `strchr`, `strcmp`. Here is a full list of standard library support for string and char manipulations.

Problem 3: Nagito Komaeda and lucky days

"I believe my actions will become the foundation of this world's hope. And... if that really happens... Praise me."

Description

Some people believe numbers with certain properties, like palindromes, are special, so does Nagito Komaeda. Komaeda wants to choose a date for himself to do some important things. As the Ultimate Lucky Student, he finds the dates he chooses are all palindrome dates. He is very interested in this kind of special dates.

We define a date `YYYYMMDD` (`Y`, `M`, `D` are digits for year, month, and date) to be a **palindrome date** if it is palindrome, that is, its digits remain the same after being reversed. For example, we can write Apr. 4th, 2022 as `20220404` and write Dec. 30th, 321 as `03211230` (We also zero-fill the year to make it exactly 8 digits). The date `03211230` is a palindrome date.

However, Komaeda has a special calendar, in which there are a total of m months in a year, and the i-th month will have d_i days. Luckily, there is no leap year (闰年) in his calendar, so there is no need to calculate whether a year is a leap year.

Nagito Komaeda wants to know the total number of palindrome dates in a certain time interval. He is so lucky that he can randomly guess out the correct answer, but you cannot. You want to write a program to calculate the number of palindrome dates.

Input Format

The input contains 4 lines.

In the first line is a number m, indicating the number of months each year in the special calendar.

In the second line are m numbers separated by spaces. The i-th number d_i indicates that the i-th month has d_i days.

In the next two lines, you will get two dates $start$ and end respectively, representing the start date and the end date, in the form of `YYYYMMDD`.

It is guaranteed that:

- $1 \leq m \leq 99, 1 \leq d_i \leq 99$
- $00010101 \leq start < end \leq 99999999$.

Output Format

You only need to output one integer, the number of palindrome dates in the range $[start, end]$ (both ends inclusive).

Samples

Sample 1

Input

```
12
31 28 31 30 31 30 31 31 30 31 30 31
20000101
20101231
```

Output

```
2
```

Explanation

There are two palindrome dates between the two dates: 20011002 and 20100102.

(In this sample, the calendar is exactly "our" calendar!)

Problem 4: Chiaki Nanami Wants to Play Games

"If you want to believe in someone... you need to overcome doubt first. Belief without doubt... is simply a lie."

Description

Chiaki Nanami, the Ultimate Gamer, now wants to play games. She has a lot of games in her game console and wants to play some of them. Nanami has marked every game with its game type (like FPS or Rogue-like), represented by an integer. If two game types are marked by the same integer, the two games have the same type and vice versa. Nanami also records the price of each game.

The Ultimate Game Player, Nanami Chiaki, also has her favorite type of games. However, she forgot which type of games is her favorite. She only knows that She loves those games so much that the money she spent on her favorite type of games is over half of the total cost of all games.

Please help Nanami to find her favorite game type.

Input Format

You will first recieve a number n, which indicates the number of games Nanami has. Then in each of the next n lines you will recieve a tuple. The tuple (s_i, p_i, t_i) in i-th line represents the **name**, **price** and **type** of the i-th game.

It's guaranteed that there will be no comma in s_i, the game names. The game types may not be consecutive (i.e. the t_is may not simply be $1, 2, 3...$).

It is guaranteed that

- $1 \leq n \leq 10^6$
- $1 \leq p_i \leq 2 \times 10^6$
- $1 \leq t_i \leq 10^6$
- $\sum_{i=1}^{n} |s_i| \leq 10^7$

Output Format

You only need to print one integer, Nanami's favorite type of games.

Samples

Sample 1

Input

```
3
(Goose Goose Duck,2,1)
(CSGO,4,2)
(Among us,3,1)
```

Output

```
1
```

Explanation

In the sample given, there are 3 games in Nanami's game console. The cost of game type 1 is $2+3=5$ and the cost of game type 2 is 4. The cost of game type 1 is over the half of the total cost $4+5=9$, so her favorite type is 1.

Challenge (Golden Strawberry):

This problem can be solved in a clever way! You don't need to record all games' types and costs into arrays, and in fact you only need very few variables! Can you solve the problem by declaring at most 10 variables (without using arrays or pointers)? Submit your solution to the additional problem 6 to win the golden strawberry award!

Hint: Use the property of the favorite games.

Problem 5: Gundham Tanaka and the Dark Devas of Destruction

"Pitiful humans... they refuse to lift their heads up for fear of doubting the authenticity of the blue sky..."

Description

Gundham Tanaka, the Ultimate Breeder, kept a lot of animals, such as hamsters and eagles. He named all his animals respectively (like his four favourite hamsters, "the Four Dark Devas of Destruction"). One day, the "Dark Devas of Destruction" (his hamsters, actually) wanted to play a game with him.

These hamsters form a matrix of r rows and c columns, but facing to different directions `L` (left), `R` (right), `U` (up), or `D` (down). Among them are some special hamsters which tanaka needs to find out.

At first, a leading hamster will hint where the first special hamster is. Whenever he finds a special hamster, it will also hint the position of the next special hamster to him. All hints are in forms like "The next special hamster is the p_i-th one in front/back/left/right of me".

The hamsters make mistake sometimes. When a hamster gives a position, if it is out of bound, or if the hamster at that position has already been identified as a special hamster previously , Tanaka will stop the game and tell his hamsters that they have made a mistake.

Now Tanaka is playing the game, please help him find all special hamsters.

Input Format

In the first line, you will receive 3 numbers r, c, q, separated by spaces, indicating the number of rows and columns of the hamster matrix, and the number of hints given by hamsters.

In the next r lines, you will receive a r-row c-column word matrix $\{d_{ij}\}(d_{ij} \in \{$ `L`, `R`, `U`, `D` $\})$. d_{ij} represents the direction of the hamster in the i-th row and j-th column, facing `L` (left), `R` (right), `U` (up), or `D` (down).

In the next line, you will receive 2 numbers $r_{leading}, c_{leading}$, indicating the row and column of the leading hamster.

In the next q lines, each line you will receive a word $w_i(w_i \in \{$ `"Left"`, `"Right"`, `"Front"`, `"Back"` $\})$ and an integer p_i, meaning that the hint is "The next hamster is the p_i-th in w_i of me".

Another thing you need to know: Hamsters are not computers, they count from 1 rather than 0.

It is guaranteed that:

- $2 \leq r, c \leq 500$
- $1 \leq q \leq 100000$

Output Format

In each line, you should print `(r, c)`, the row and column of a special hamster, in their order of being found, until the game ends or the last one is found. If the hamsters made mistake in one hint, you should print `Mistake!` in the corresponding line and end the game.

Samples

Sample 1

Input

```
2 2 4
U L
D D
1 1
Back 1
Left 1
Left 10
Right 1
```

Output

```
(2, 1)
(2, 2)
Mistake!
```

Explanation

Their directions:

↑ ←
↓ ↓

The leading hamster is at $(1, 1)$. It says that the first special hamster is the 1st hamster in its back, at $(2, 1)$.

Notice that the hamster at $(2, 1)$ is facing down, and it says the next special hamster is in **its** left, which actually means to the right (back's left) of the matrix, which is $(2, 2)$.

The next special hamster is obviously unable to find, so there is a mistake. The game ends, ignoring any more hints from hamsters.