Codeforces Beta Round #57 (Div. 2)

A. Ultra-Fast Mathematician

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Shapur was an extremely gifted student. He was great at everything including Combinatorics, Algebra, Number Theory, Geometry, Calculus, etc. He was not only smart but extraordinarily fast! He could manage to sum 10^{18} numbers in a single second.

One day in 230 AD Shapur was trying to find out if any one can possibly do calculations faster than him. As a result he made a very great contest and asked every one to come and take part.

In his contest he gave the contestants many different pairs of numbers. Each number is made from digits 0 or 1. The contestants should write a new number corresponding to the given pair of numbers. The rule is simple: The i-th digit of the answer is 1 if and only if the i-th digit of the two given numbers differ. In the other case the i-th digit of the answer is 0.

Shapur made many numbers and first tried his own speed. He saw that he can perform these operations on numbers of length ∞ (length of a number is number of digits in it) in a glance! He always gives correct answers so he expects the contestants to give correct answers, too. He is a good fellow so he won't give anyone very big numbers and he always gives one person numbers of same length.

Now you are going to take part in Shapur's contest. See if you are faster and more accurate.

Input

There are two lines in each input. Each of them contains a single number. It is guaranteed that the numbers are made from 0 and 1 only and that their length is same. The numbers may start with 0. The length of each number doesn't exceed 100.

Output

Write one line — the corresponding answer. Do not omit the leading 0s.

Sample test(s)

input
1010100 0100101
output
1110001

input
000 111
output
111

input
1110 1010
output
0100

input
01110 01100
output
00010

B. Hard Work

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After the contest in comparing numbers, Shapur's teacher found out that he is a real genius and that no one could possibly do the calculations faster than him even using a super computer!

Some days before the contest, the teacher took a very simple-looking exam and all his n students took part in the exam. The teacher gave them 3 strings and asked them to *concatenate* them. Concatenating strings means to put them in some arbitrary order one after the other. For example from concatenating `Alireza` and `Amir` we can get to `AlirezaAmir` or `AmirAlireza` depending on the order of concatenation.

Unfortunately enough, the teacher forgot to ask students to concatenate their strings in a pre-defined order so each student did it the way he/she liked.

Now the teacher knows that Shapur is such a fast-calculating genius boy and asks him to correct the students' papers.

Shapur is not good at doing such a time-taking task. He rather likes to finish up with it as soon as possible and take his time to solve 3-SAT in polynomial time. Moreover, the teacher has given some advice that Shapur has to follow. Here's what the teacher said:

- As I expect you know, the strings I gave to my students (including you) contained only lowercase and uppercase Persian Mikhi-Script letters. These letters are too much like Latin letters, so to make your task much harder I converted all the initial strings and all of the students' answers to Latin.
- As latin alphabet has much less characters than Mikhi-Script, I added three odd-looking characters to the answers, these include "`-`", "`;`" and "`_`". These characters are my own invention of course! And I call them *Signs*.
- The length of all initial strings was less than or equal to 100 and the lengths of my students' answers are less than or equal to 600
- My son, not all students are genius as you are. It is quite possible that they make minor mistakes changing case of some characters. For example they may write `ALiReZaAmIR` instead of `AlirezaAmir`. Don't be picky and ignore these mistakes.
- Those signs which I previously talked to you about are not important. You can ignore them, since many students are in the mood for adding extra signs or forgetting about a sign. So something like `Iran;;--` is the same as `--;IRAN`
- You should indicate for any of my students if his answer was right or wrong. Do this by writing "`WA`" for Wrong answer or "`ACC`" for a correct answer.
- I should remind you that none of the strings (initial strings or answers) are empty.
- Finally, do these as soon as possible. You have less than 2 hours to complete this.

Input

The first three lines contain a string each. These are the initial strings. They consists only of lowercase and uppercase Latin letters and signs (" `-`", "`;`" and "`_`"). All the initial strings have length from 1 to 100, inclusively.

In the fourth line there is a single integer n ($0 \le n \le 1000$), the number of students.

Next n lines contain a student's answer each. It is guaranteed that the answer meets what the teacher said. Each answer iconsists only of lowercase and uppercase Latin letters and signs ("`-`", "`;`" and "`_`"). Length is from 1 to 600, inclusively.

Output

For each student write in a different line. Print "`WA`" if his answer is wrong or "`ACC`" if his answer is OK.

Sample test(s)

input

```
Iran_
Persian;
W_o;n;d;e;r;f;u;l;
7
WonderfulPersianIran
wonderful_PersIAN_IRAN;;_
WONDERFUL___IRAN__PERSIAN__;;
Ira__Persiann__Wonderful
Wonder;;fulPersian___;I;r;a;n;
__________IranPersianWonderful__________
PersianIran_is_Wonderful
```

output

```
ACC
ACC
ACC
WA
ACC
ACC
WA
```

input

```
Shapur;;
is___
a_genius
```

```
3
Shapur__a_is___geniUs
is___shapur___a__Genius;
Shapur;;is;;a;;geni;;us;;
```

output

```
WA
ACC
ACC
```

C. Capture Valerian

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

It's now 260 AD. Shapur, being extremely smart, became the King of Persia. He is now called Shapur, His majesty King of kings of Iran and Aniran.

Recently the Romans declared war on Persia. They dreamed to occupy Armenia. In the recent war, the Romans were badly defeated. Now their senior army general, Philip is captured by Shapur and Shapur is now going to capture Valerian, the Roman emperor.

Being defeated, the cowardly Valerian hid in a room at the top of one of his castles. To capture him, Shapur has to open many doors. Fortunately Valerian was too scared to make impenetrable locks for the doors.

Each door has 4 parts. The first part is an integer number a. The second part is either an integer number b or some really odd sign which looks like `R`. The third one is an integer c and the fourth part is empty! As if it was laid for writing something. Being extremely gifted, after opening the first few doors, Shapur found out the secret behind the locks.

c is an integer written in base a, to open the door we should write it in base b. The only bad news is that this `R` is some sort of special numbering system that is used only in Roman empire, so opening the doors is not just a piece of cake!

Here's an explanation of this really weird number system that even doesn't have zero:

Roman numerals are based on seven symbols: a stroke (identified with the letter `I`) for a unit, a chevron (identified with the letter `V`) for a five, a cross-stroke (identified with the letter `X`) for a ten, a `C` (identified as an abbreviation of Centum) for a hundred, etc.:

- `I`=1
- `V`=5
- `X`=10
- `L`=50
- `C`=100
- `D`=500
- `M`=1000

Symbols are iterated to produce multiples of the decimal (1, 10, 100, $1,000$) values, with `V`, `L`, `D` substituted for a multiple of five, and the iteration continuing: `I` 1, `II` 2, `III` 3, `V` 5, `VI` 6, `VII` 7, etc., and the same for other bases: `X` 10, `XX` 20, `XXX` 30, `L` 50, `LXXX` 80; `CC` 200, `DCC` 700, etc. At the fourth and ninth iteration, a subtractive principle must be employed, with the base placed before the higher base: `IV` 4, `IX` 9, `XL` 40, `XC` 90, `CD` 400, `CM` 900.

Also in bases greater than 10 we use `A` for 10, `B` for 11, etc.

Help Shapur capture Valerian and bring peace back to Persia, especially Armenia.

Input

The first line contains two integers a and b ($2 \le a, b \le 25$). Only b may be replaced by an `R` which indicates Roman numbering system.

The next line contains a single non-negative integer c in base a which may contain leading zeros but its length doesn't exceed 10^3.

It is guaranteed that if we have Roman numerals included the number would be less than or equal to 3000_{10} and it won't be 0. In any other case the number won't be greater than $10^{15}{}_{10}$.

Output

Write a single line that contains integer c in base b. You must omit leading zeros.

Sample test(s)

input
10 2 1
output
1

input
16 R 5
output
V

input
5 R 4

output

```
IV
```

input

```
2 2
1111001
```

output

```
1111001
```

input

```
12 13
A
```

output

```
A
```

Note

You can find more information about roman numerals here: http://en.wikipedia.org/wiki/Roman_numerals

D. Eternal Victory

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Valerian was captured by Shapur. The victory was such a great one that Shapur decided to carve a scene of Valerian's defeat on a mountain. So he had to find the best place to make his victory eternal!

He decided to visit all n cities of Persia to find the best available mountain, but after the recent war he was too tired and didn't want to traverse a lot. So he wanted to visit each of these n cities at least once with smallest possible traverse. Persian cities are connected with bidirectional roads. You can go from any city to any other one using these roads and there is a unique path between each two cities.

All cities are numbered 1 to n. Shapur is currently in the city 1 and he wants to visit all other cities with minimum possible traverse. He can finish his travels in any city.

Help Shapur find how much He should travel.

Input

First line contains a single natural number n ($1 \le n \le 10^5$) — the amount of cities.

Next n - 1 lines contain 3 integer numbers each x_i, y_i and w_i ($1 \le x_i, y_i \le n, 0 \le w_i \le 2 \times 10^4$). x_i and y_i are two ends of a road and w_i is the length of that road.

Output

A single integer number, the minimal length of Shapur's travel.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cout` (also you may use `%I64d`).

Sample test(s)

input
3 1 2 3 2 3 4
output
7

input
3 1 2 3 1 3 3
output
9

E. Enemy is weak

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Romans have attacked again. This time they are much more than the Persians but Shapur is ready to defeat them. He says: "A lion is never afraid of a hundred sheep".

Nevertheless Shapur has to find weaknesses in the Roman army to defeat them. So he gives the army a weakness number.

In Shapur's opinion the weakness of an army is equal to the number of triplets i, j, k such that $i < j < k$ and $a_i > a_j > a_k$ where a_x is the power of man standing at position x. The Roman army has one special trait — powers of all the people in it are distinct.

Help Shapur find out how weak the Romans are.

Input

The first line of input contains a single number n ($3 \le n \le 10^6$) — the number of men in Roman army. Next line contains n different positive integers a_i ($1 \le i \le n, 1 \le a_i \le 10^9$) — powers of men in the Roman army.

Output

A single integer number, the weakness of the Roman army.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cout` (also you may use `%I64d`).

Sample test(s)

input
3 3 2 1

output
1

input
3 2 3 1

output
0

input
4 10 8 3 1

output
4

input
4 1 5 4 3

output
1