Problem A: $\mathbf{Con} + \cfrac{\mathbf{tin}}{\mathbf{ued} + \cfrac{\mathbf{Frac}}{\mathbf{tions}}}$

The (simple) continued fraction representation of a real number r is an expression obtained by an iterative process of representing r as a sum of its integer part and the reciprocal of another number, then writing this other number as the sum of its integer part and another reciprocal, and so on. In other words, a continued fraction representation of r is of the form

$$r = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \dots}}}$$

where $a_0, a_1, a_2, \dots$ are integers and $a_1, a_2, \dots > 0$. We call the a_i-values *partial quotients*. For example, in the continued fraction representation of 5.4 the partial quotients are $a_0 = 5, a_1 = 2$, and $a_2 = 2$. This representation of a real number has several applications in theory and practice.

While irrational numbers like $\sqrt{2}$ require an infinite set of partial quotients, any rational number can be written as a continued fraction with a unique, finite set of partial quotients (where the last partial quotient is never 1 in order to preserve uniqueness). Given two rational numbers in continued fraction representation, your task is to perform the four elementary arithmetic operations on these numbers and display the result in continued fraction representation.

Input Time Limit: 3 secs, No. of Test Cases: 500, Input File Size 13.1K

Each test case consists of three lines. The first line contains two integers n_1 and n_2, $1 \le n_i \le 9$ specifying the number of partial quotients of two rational numbers r_1 and r_2. The second line contains the partial quotients of r_1 and the third line contains the partial quotients of r_2. The partial quotients satisfy $|a_0| \le 10$ and $0 < a_i \le 10$, the last partial quotient will never be 1, and r_2 is non-zero. A line containing two 0's will terminate input.

Output

For each test case, display the case number followed by the continued fraction representation of $r_1 + r_2$, $r_1 - r_2$, $r_1 \times r_2$, and r_1 / r_2 in order, each on a separate line. Use 64-bit integers for all of your calculations (`long long` in C++ and `long` in Java).

Sample Input

```
4 3
5 1 1 2
5 2 2
0 0
```

Sample Output

```
Case 1:
11
0 5
30 4 6
1 27
```

Problem B Time Warp

Tim Ang is a bit of a nerd. Check that – he's a HUGE nerd. When you ask him the time, he might say something like "20 after 8", which seems normal, but other times he'll say things like "90 after 8" or "126 til 4", which gives you pause. When you ask him about this, Tim say that "20 after 8" means the first time after 8 that the hour and minute hands of the clock make an angle of 20 degrees; "126 til 4" means the closest time before 4 that the hands make an angle of 126 degrees. As Tim walks away snickering, you resolve that you will write a program that will automatically convert Tim's times to our more normal, non-nerdy times. That'll show the little geek!

Input **Time Limit: 3 secs, No. of Test Cases: 8640, Input File Size 94.6K**

The input file starts with an integer n indicating the number of test cases. Each test case consists of a single line of the form a `after` b or a `til` b, where a and b are integers satisfying $0 \leq a < 360$, and $1 \leq b \leq 12$. All angles are measured starting at the hour hand and moving clockwise until reaching the minute hand (so, for example, at 9 o'clock the hands make an angle of 90 degrees and at 3 o'clock they make an angle of 270).

Output

For each test case, output the time in the format `h:m:s`, where `h`, `m` and `s` are the hour, minutes and seconds closest to the given angle and hour and $1 \leq$ `h` ≤ 12. Always use two digits to represent the number of minutes and seconds.

Sample Input

```
4
20 after 8
126 til 4
180 til 1
0 after 12
```

Sample Output

```
Case 1: 8:47:16
Case 2: 3:39:16
Case 3: 12:32:44
Case 4: 1:05:27
```

Problem C   A Cure for the Common Code

You've been tasked with relaying coded messages to your fellow resistance fighters. Each coded message is a sequence of lower-case letters that you furtively scrawl on monuments in the dead of night.

Since you're writing these messages by hand, the longer the message, the greater the likelihood of being caught by the evil empire while writing. Because of this you decide it would be worthwhile to come up with a simple encoding that might allow for shorter messages. After thinking about it for a while, you decide to use integers and parentheses to indicate repetition of substrings when doing so shortens the number of characters you need to write. For example, the 10 character string

abcbcbcbca

could be more briefly written as the 7 character string

a4(bc)a

If a single letter is being repeated, parentheses are not needed. Also, repetitions may themselves be repeated, so you can write the 20 character string

abbbcdcdcdabbbcdcdcd

as the 11 character string

2(a3b3(cd))

and so forth.

Input   Time Limit: 5 secs, No. of Test Cases: 39, Input File Size 2.95K

Each test case consists of a single line containing a string of lower-case letters of length ≤ 500. A line containing a single 0 will terminate the input.

Output

For each test case, output the number of characters needed for a minimal encoding of the string.

Sample Input

```
abcbcbcbca
abbbcdcdcdabbbcdcdcd
0
```

Sample Output

```
Case 1: 7
Case 2: 11
```

D. Lonely Dreamoon 2

time limit per test: 1 second
memory limit per test: 512 mebibytes
input: *standard input*
output: *standard output*

Dreamoon, who doesn't have a girlfriend, often goes for a walk along some streets in Taipei while thinking about problems from algorithm competitions. Unfortunately, there are so many couples acting lovey-dovey on the street that Dreamoon can not focus on thinking about those problems.

One day, despite the love birds everywhere, Dreamoon discovered a problem input containing an integer sequence: $a_1, a_2, a_3, ..., a_N$.

Dreamoon thought: because I'm single, every pair of consecutive numbers should have a large difference! This is, Dreamoon wants to reorder the sequence to make the value $\min_{i=2...N}(|a_i - a_{i-1}|)$ as large as possible.

So Dreamoon turned on Drazil, who does have a girlfriend, and forced Drazil to fulfill the above condition by reordering the integer sequence. Please help poor Drazil $>\_<$

Input

The input consists of two lines. The first line contains an integer N. The second line consists of N integers $a_1, a_2, ..., a_N$.

- $2 \le N \le 2 \times 10^5$
- $-10^9 \le a_i \le 10^9$

Output

Output a single line consisting of N integers, denoting the integer sequence a after reordering. For this reordering, the value $\min_{i=2...N} |a_i - a_{i-1}|$ must be the largest possible among all reorderings of the input sequence. If there are several possible answers, output any one of them.

Examples

input

```
3
3 1 5
```

output

```
3 5 1
```

input

```
4
-1 -1 1 1
```

output

```
1 -1 1 -1
```

E. Forest Game

time limit per test: 4 seconds
memory limit per test: 512 mebibytes
input: *standard input*
output: *standard output*

Consider the following boring game about removing nodes from a forest. Initially, the forest contains only one tree with N nodes, and your initial score is 0. The game then goes as follows:

1. If the forest is empty, the game is finished. Otherwise, you choose one node from the current forest uniformly at random.
2. Your score increases by the size of the tree which your chosen node belongs to.
3. Remove your chosen node and all edges connected to this node. Then proceed to step 1.

Please calculate the expected value of your final score multiplied by $N!$, modulo $10^9 + 7$.

Input

The first line of input contains one integer N indicating the number of nodes in the initial tree.

Each of the following $N - 1$ lines contains two integers x and y, indicating that x-th node and y-th node are connected by an edge in the given tree. The nodes are numbered from 1 to N.

- $1 \leq N \leq 10^5$
- $1 \leq x, y \leq N$
- the given graph is a tree

Output

Output one number: the expected value of the final score of this boring game multiplied by $N!$, modulo $10^9 + 7$.

Examples

input

```
2
1 2
```

output

```
6
```

input

```
3
1 2
2 3
```

output

```
34
```

F. Tree Game

time limit per test: 1 second
memory limit per test: 512 mebibytes
input: *standard input*
output: *standard output*

Consider the following game about coloring edges in a tree.

You are given a tree. Initially, the color of all edges is white. Let a *valid path* be a simple path such that all its edges are white, and the two endpoints are leaves in the tree. On each step of this game, you can choose a valid path and paint all its edges black. You cannot stop your game until you cannot find any valid path.

The purpose of this game is to use the minimum number of steps to complete the game. Please find the minimum number of steps for the given tree.

Input

The first line of input contains one integer N indicating the number of nodes in the given tree.

Each of the following $N - 1$ lines contains two integers x and y indicating that x-th node and y-th node are connected by an edge in the given tree. Nodes are numbered from 1 to N.

- $2 \le N \le 10^5$
- $1 \le x, y \le N$
- the given graph is a tree

Output

Output one integer: the minimum number of steps required to complete the game on the given tree.

Examples

input

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

output

```
1
```

input

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

output

```
3
```