CROC-MBTU 2012, Elimination Round (ACM-ICPC)

A. System Administrator

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

Polycarpus is a system administrator. There are two servers under his strict guidance — a and b. To stay informed about the servers' performance, Polycarpus executes commands "`ping a`" and "`ping b`". Each ping command sends exactly ten packets to the server specified in the argument of the command. Executing a program results in two integers x and y ($x + y = 10$; $x, y \geq 0$). These numbers mean that x packets successfully reached the corresponding server through the network and y packets were lost.

Today Polycarpus has performed overall n ping commands during his workday. Now for each server Polycarpus wants to know whether the server is "alive" or not. Polycarpus thinks that the server is "alive", if at least half of the packets that we send to this server reached it successfully along the network.

Help Polycarpus, determine for each server, whether it is "alive" or not by the given commands and their results.

Input

The first line contains a single integer n ($2 \leq n \leq 1000$) — the number of commands Polycarpus has fulfilled. Each of the following n lines contains three integers — the description of the commands. The i-th of these lines contains three space-separated integers t_i, x_i, y_i ($1 \leq t_i \leq 2$; $x_i, y_i \geq 0$; $x_i + y_i = 10$). If $t_i = 1$, then the i-th command is "`ping a`", otherwise the i-th command is "`ping b`". Numbers x_i, y_i represent the result of executing this command, that is, x_i packets reached the corresponding server successfully and y_i packets were lost.

It is guaranteed that the input has at least one "`ping a`" command and at least one "`ping b`" command.

Output

In the first line print string "`LIVE`" (without the quotes) if server a is "alive", otherwise print "`DEAD`" (without the quotes).

In the second line print the state of server b in the similar format.

Examples

input
2 1 5 5 2 6 4
output
LIVE LIVE

input
3 1 0 10 2 0 10 1 10 0
output
LIVE DEAD

Note

Consider the first test case. There 10 packets were sent to server a, 5 of them reached it. Therefore, at least half of all packets sent to this server successfully reached it through the network. Overall there were 10 packets sent to server b, 6 of them reached it. Therefore, at least half of all packets sent to this server successfully reached it through the network.

Consider the second test case. There were overall 20 packages sent to server a, 10 of them reached it. Therefore, at least half of all packets sent to this server successfully reached it through the network. Overall 10 packets were sent to server b, 0 of them reached it. Therefore, less than half of all packets sent to this server successfully reached it through the network.

B. Internet Address

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

Vasya is an active Internet user. One day he came across an Internet resource he liked, so he wrote its address in the notebook. We know that the address of the written resource has format:

<protocol>://<domain>.ru[/<context>]

where:

- <protocol> can equal either "http" (without the quotes) or "ftp" (without the quotes),
- <domain> is a non-empty string, consisting of lowercase English letters,
- the /<context> part may not be present. If it is present, then <context> is a non-empty string, consisting of lowercase English letters.

If string <context> isn't present in the address, then the additional character "/" isn't written. Thus, the address has either two characters "/" (the ones that go before the domain), or three (an extra one in front of the context).

When the boy came home, he found out that the address he wrote in his notebook had no punctuation marks. Vasya must have been in a lot of hurry and didn't write characters ":", "/", ".".

Help Vasya to restore the possible address of the recorded Internet resource.

Input

The first line contains a non-empty string that Vasya wrote out in his notebook. This line consists of lowercase English letters only.

It is guaranteed that the given string contains at most 50 letters. It is guaranteed that the given string can be obtained from some correct Internet resource address, described above.

Output

Print a single line — the address of the Internet resource that Vasya liked. If there are several addresses that meet the problem limitations, you are allowed to print any of them.

Examples

input
httpsunrux
output
http://sun.ru/x

input
ftphttprururu
output
ftp://http.ru/ruru

Note

In the second sample there are two more possible answers: "ftp://httpruru.ru" and "ftp://httpru.ru/ru".

C. Game with Coins

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

Two pirates Polycarpus and Vasily play a very interesting game. They have n chests with coins, the chests are numbered with integers from 1 to n. Chest number i has a_i coins.

Polycarpus and Vasily move in turns. Polycarpus moves first. During a move a player is allowed to choose a positive integer x $(2 \cdot x + 1 \le n)$ and take a coin from each chest with numbers x, $2 \cdot x$, $2 \cdot x + 1$. It may turn out that some chest has no coins, in this case the player doesn't take a coin from this chest. The game finishes when all chests get emptied.

Polycarpus isn't a greedy scrooge. Polycarpys is a lazy slob. So he wonders in what minimum number of moves the game can finish. Help Polycarpus, determine the minimum number of moves in which the game can finish. Note that Polycarpus counts not only his moves, he also counts Vasily's moves.

Input

The first line contains a single integer n $(1 \le n \le 100)$ — the number of chests with coins. The second line contains a sequence of space-separated integers: $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 1000)$, where a_i is the number of coins in the chest number i at the beginning of the game.

Output

Print a single integer — the minimum number of moves needed to finish the game. If no sequence of turns leads to finishing the game, print -1.

Examples

input
1 1
output
-1

input
3 1 2 3
output
3

Note

In the first test case there isn't a single move that can be made. That's why the players won't be able to empty the chests.

In the second sample there is only one possible move $x = 1$. This move should be repeated at least 3 times to empty the third chest.

D. Restoring Table

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

Recently Polycarpus has learned the "bitwise AND" operation (which is also called "AND") of non-negative integers. Now he wants to demonstrate the school IT teacher his superb manipulation with the learned operation.

For that Polycarpus came to school a little earlier and wrote on the board a sequence of non-negative integers $a_1, a_2, \ldots, a_n$. He also wrote a square matrix b of size $n \times n$. The element of matrix b that sits in the i-th row in the j-th column (we'll denote it as b_{ij}) equals:

- the "bitwise AND" of numbers a_i and a_j (that is, $b_{ij} = a_i \ \& \ a_j$), if $i \neq j$;
- -1, if $i = j$.

Having written out matrix b, Polycarpus got very happy and wiped a off the blackboard. But the thing is, the teacher will want this sequence to check whether Polycarpus' calculations were correct. Polycarus urgently needs to restore the removed sequence of integers, or else he won't prove that he can count correctly.

Help Polycarpus, given matrix b, restore the sequence of numbers $a_1, a_2, \ldots, a_n$, that he has removed from the board. Polycarpus doesn't like large numbers, so any number in the restored sequence mustn't exceed 10^9.

Input

The first line contains a single integer n $(1 \le n \le 100)$ — the size of square matrix b. Next n lines contain matrix b. The i-th of these lines contains n space-separated integers: the j-th number represents the element of matrix b_{ij}. It is guaranteed, that for all i $(1 \le i \le n)$ the following condition fulfills: $b_{ii} = -1$. It is guaranteed that for all i, j $(1 \le i, j \le n;\ i \neq j)$ the following condition fulfills: $0 \le b_{ij} \le 10^9$, $b_{ij} = b_{ji}$.

Output

Print n non-negative integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$ — the sequence that Polycarpus wiped off the board. Separate the numbers by whitespaces.

It is guaranteed that there is sequence a that satisfies the problem conditions. If there are multiple such sequences, you are allowed to print any of them.

Examples

input
1 -1
output
0

input
3 -1 18 0 18 -1 0 0 0 -1
output
18 18 0

input
4 -1 128 128 128 128 -1 148 160 128 148 -1 128 128 160 128 -1
output
128 180 148 160

Note

If you do not know what is the "bitwise AND" operation please read: `http://en.wikipedia.org/wiki/Bitwise_operation`.

E. Mishap in Club

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

Polycarpus just has been out of luck lately! As soon as he found a job in the "Binary Cat" cafe, the club got burgled. All ice-cream was stolen.

On the burglary night Polycarpus kept a careful record of all club visitors. Each time a visitor entered the club, Polycarpus put down character "+" in his notes. Similarly, each time a visitor left the club, Polycarpus put character "-" in his notes. We know that all cases of going in and out happened consecutively, that is, no two events happened at the same time. Polycarpus doesn't remember whether there was somebody in the club at the moment when his shift begun and at the moment when it ended.

Right now the police wonders what minimum number of distinct people Polycarpus could have seen. Assume that he sees anybody coming in or out of the club. Each person could have come in or out an arbitrary number of times.

Input

The only line of the input contains a sequence of characters "+" and "-", the characters are written one after another without any separators. The characters are written in the order, in which the corresponding events occurred. The given sequence has length from 1 to 300 characters, inclusive.

Output

Print the sought minimum number of people

Examples

input
+-+-+
output
1

input
output
3

F. Log Stream Analysis

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

You've got a list of program warning logs. Each record of a log stream is a string in this format:

"2012-MM-DD HH:MM:SS:MESSAGE" (without the quotes).

String "MESSAGE" consists of spaces, uppercase and lowercase English letters and characters "!", ".", ",", "?". String "2012-MM-DD" determines a correct date in the year of 2012. String "HH:MM:SS" determines a correct time in the 24 hour format.

The described record of a log stream means that at a certain time the record has got some program warning (string "MESSAGE" contains the warning's description).

Your task is to print the first moment of time, when the number of warnings for the last n seconds was not less than m.

Input

The first line of the input contains two space-separated integers n and m ($1 \le n, m \le 10000$).

The second and the remaining lines of the input represent the log stream. The second line of the input contains the first record of the log stream, the third line contains the second record and so on. Each record of the log stream has the above described format. All records are given in the chronological order, that is, the warning records are given in the order, in which the warnings appeared in the program.

It is guaranteed that the log has at least one record. It is guaranteed that the total length of all lines of the log stream doesn't exceed $5 \cdot 10^6$ (in particular, this means that the length of some line does not exceed $5 \cdot 10^6$ characters). It is guaranteed that all given dates and times are correct, and the string 'MESSAGE" in all records is non-empty.

Output

If there is no sought moment of time, print -1. Otherwise print a string in the format "2012-MM-DD HH:MM:SS" (without the quotes) — the first moment of time when the number of warnings for the last n seconds got no less than m.

Examples

input
60 3 2012-03-16 16:15:25: Disk size is 2012-03-16 16:15:25: Network failute 2012-03-16 16:16:29: Cant write varlog 2012-03-16 16:16:42: Unable to start process 2012-03-16 16:16:43: Disk size is too small 2012-03-16 16:16:53: Timeout detected
output
2012-03-16 16:16:43

input
1 2 2012-03-16 23:59:59:Disk size 2012-03-17 00:00:00: Network 2012-03-17 00:00:01:Cant write varlog
output
-1

input
2 2 2012-03-16 23:59:59:Disk size is too sm 2012-03-17 00:00:00:Network failute dete 2012-03-17 00:00:01:Cant write varlogmysq
output
2012-03-17 00:00:00

G. Suggested Friends

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

Polycarpus works as a programmer in a start-up social network. His boss gave his a task to develop a mechanism for determining suggested friends. Polycarpus thought much about the task and came to the folowing conclusion.

Let's say that all friendship relationships in a social network are given as m username pairs a_i, b_i $(a_i \neq b_i)$. Each pair a_i, b_i means that users a_i and b_i are friends. Friendship is symmetric, that is, if a_i is friends with b_i, then b_i is also friends with a_i. User y is a *suggested friend* for user x, if the following conditions are met:

1. $x \neq y$;
2. x and y aren't friends;
3. among all network users who meet the first two conditions, user y has most of all common friends with user x. User z is a common friend of user x and user y $(z \neq x, z \neq y)$, if x and z are friends, and y and z are also friends.

Your task is to help Polycarpus to implement a mechanism for determining suggested friends.

Input

The first line contains a single integer m $(1 \leq m \leq 5000)$ — the number of pairs of friends in the social network. Next m lines contain pairs of names of the users who are friends with each other. The i-th line contains two space-separated names a_i and b_i $(a_i \neq b_i)$. The users' names are non-empty and consist of at most 20 uppercase and lowercase English letters.

It is guaranteed that each pair of friends occurs only once in the input. For example, the input can't contain x, y and y, x at the same time. It is guaranteed that distinct users have distinct names. It is guaranteed that each social network user has at least one friend. The last thing guarantees that each username occurs at least once in the input.

Output

In the first line print a single integer n — the number of network users. In next n lines print the number of suggested friends for each user. In the i-th line print the name of the user c_i and the number of his suggested friends d_i after a space.

You can print information about the users in any order.

Examples

input
5 Mike Gerald Kate Mike Kate Tank Gerald Tank Gerald David
output
5 Mike 1 Gerald 1 Kate 1 Tank 1 David 2

input
4 valera vanya valera edik pasha valera igor valera
output
5 valera 0 vanya 3 edik 3 pasha 3 igor 3

Note

In the first test case consider user David. Users Mike and Tank have one common friend (Gerald) with David. User Kate has no common friends with David. That's why David's suggested friends are users Mike and Tank.

H. Queries for Number of Palindromes

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

You've got a string $s = s_1 s_2 \ldots s_{|s|}$ of length $|s|$, consisting of lowercase English letters. There also are q queries, each query is described by two integers l_i, r_i $(1 \le l_i \le r_i \le |s|)$. The answer to the query is the number of substrings of string $s[l_i \ldots r_i]$, which are palindromes.

String $s[l \ldots r] = s_l s_{l+1} \ldots s_r$ $(1 \le l \le r \le |s|)$ is a *substring* of string $s = s_1 s_2 \ldots s_{|s|}$.

String t is called a *palindrome*, if it reads the same from left to right and from right to left. Formally, if $t = t_1 t_2 \ldots t_{|t|} = t_{|t|} t_{|t|-1} \ldots t_1$.

Input

The first line contains string s $(1 \le |s| \le 5000)$. The second line contains a single integer q $(1 \le q \le 10^6)$ — the number of queries. Next q lines contain the queries. The i-th of these lines contains two space-separated integers l_i, r_i $(1 \le l_i \le r_i \le |s|)$ — the description of the i-th query.

It is guaranteed that the given string consists only of lowercase English letters.

Output

Print q integers — the answers to the queries. Print the answers in the order, in which the queries are given in the input. Separate the printed numbers by whitespaces.

Examples

input
caaaba 5 1 1 1 4 2 3 4 6 4 5

output
1 7 3 4 2

Note

Consider the fourth query in the first test case. String $s[4 \ldots 6]$ = «aba». Its palindrome substrings are: «a», «b», «a», «aba».