D. A Simple Task

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

Given a simple graph, output the number of simple cycles in it. A simple cycle is a cycle with no repeated vertices or edges.

Input

The first line of input contains two integers n and m ($1 \le n \le 19$, $0 \le m$) – respectively the number of vertices and edges of the graph. Each of the subsequent m lines contains two integers a and b, ($1 \le a, b \le n$, $a \ne b$) indicating that vertices a and b are connected by an undirected edge. There is no more than one edge connecting any pair of vertices.

Output

Output the number of cycles in the given graph.

Examples

input
4 6 1 2 1 3 1 4 2 3 2 4 3 4
output
7

Note

The example graph is a clique and contains four cycles of length 3 and three cycles of length 4.

A. Train and Peter

time limit per test
1 second
memory limit per test
64 megabytes
input
standard input
output
standard output

Peter likes to travel by train. He likes it so much that on the train he falls asleep.
Once in summer Peter was going by train from city A to city B, and as usual, was sleeping. Then he woke up, started to look through the window and noticed that every railway station has a flag of a particular colour.
The boy started to memorize the order of the flags' colours that he had seen. But soon he fell asleep again. Unfortunately, he didn't sleep long, he woke up and went on memorizing the colours. Then he fell asleep again, and that time he slept till the end of the journey.
At the station he told his parents about what he was doing, and wrote two sequences of the colours that he had seen before and after his sleep, respectively.
Peter's parents know that their son likes to fantasize. They give you the list of the flags' colours at the stations that the train passes sequentially on the way from A to B, and ask you to find out if Peter could see those sequences on the way from A to B, or from B to A. Remember, please, that Peter had two periods of wakefulness.
Peter's parents put lowercase Latin letters for colours. The same letter stands for the same colour, different letters — for different colours.

Input
The input data contains three lines. The first line contains a non-empty string, whose length does not exceed 10^5,

the string consists of lowercase Latin letters — the flags' colours at the stations on the way from A to B. On the way from B to A the train passes the same stations, but in reverse order.
The second line contains the sequence, written by Peter during the first period of wakefulness. The third line contains the sequence, written during the second period of wakefulness. Both sequences are non-empty, consist of lowercase Latin letters, and the length of each does not exceed 100 letters. Each of the sequences is written in chronological order.

Output
Output one of the four words without inverted commas:

- «forward» — if Peter could see such sequences only on the way from A to B;
- «backward» — if Peter could see such sequences on the way from B to A;
- «both» — if Peter could see such sequences both on the way from A to B, and on the way from B to A;
- «fantasy» — if Peter could not see such sequences.

Examples

input
atob a b
output
forward

input
aaacaaa aca aa
output
both

Note
It is assumed that the train moves all the time, so one flag cannot be seen twice. There are no flags at stations A and B.

A. Almost Prime

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

A number is called almost prime if it has exactly two distinct prime divisors. For example, numbers 6, 18, 24 are almost prime, while 4, 8, 9, 42 are not. Find the amount of almost prime numbers which are between 1 and n, inclusive.

Input

Input contains one integer number n ($1 \le n \le 3000$).

Output

Output the amount of almost prime numbers between 1 and n, inclusive.

Examples

input
10
output
2

input
21
output
8

C. Dijkstra?

time limit per test: 1 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given a weighted undirected graph. The vertices are enumerated from 1 to n. Your task is to find the shortest path between the vertex 1 and the vertex n.

Input

The first line contains two integers n and m ($2 \le n \le 10^5, 0 \le m \le 10^5$), where n is the number of vertices and m is the number of edges. Following m lines contain one edge each in form a_i, b_i and w_i ($1 \le a_i, b_i \le n, 1 \le w_i \le 10^6$), where a_i, b_i are edge endpoints and w_i is the length of the edge.

It is possible that the graph has loops and multiple edges between pair of vertices.

Output

Write the only integer `-1` in case of no path. Write the shortest path in opposite case. If there are many solutions, print any of them.

Examples

input
5 6 1 2 2 2 5 5 2 3 4 1 4 1 4 3 3 3 5 1
output
1 4 3 5

input
5 6 1 2 2 2 5 5 2 3 4 1 4 1 4 3 3 3 5 1
output
1 4 3 5

H. Reverse It!

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The 14th of March was the international day of mathematics, because of number $\pi = 3.1415926...$

In the occasion of this day Goofy Nephews Unity Organization (GNU) wants to publish the fastest program in math at 1:59:26 AM.

Now the time is 1:11:11 AM and the project team haven't checked their program yet. Because of shortage of time they want to check their program with some queries. So they hired Hormizd (one of the greatest programmers in world) to write a tester for GNU's new program. Because Hormizd has much more important things to do, he wants you to write a small part of tester and it is reversing the numbers. Help him before 1:59:26.

We can reverse numbers easily. For example by reversing 1234 we get 4321.

Note, that if the integer is negative then its reverse would be also negative. For example reverse of -123 is -321.

Also, you have to delete all the leading zeroes before and after the reverse.

Given an integer you have to help Hormizd reverse it.

Input

The first line contains a single integer n. It is less than 10^{1000} by it's absolute value. This integer may have leading zeros. If it had leading zeros you should first omit them and print the reverse of remaining digits. It's guaranteed that input contains less than 10001 characters.

Output

Output a single integer, the reverse of the given number. You have to omit leading zeros in the output.

Examples

input
23
output
32

input
-032
output
-23

input
01234560
output
654321

B. Equation

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an equation:

$$Ax^2 + Bx + C = 0.$$

Your task is to find the number of distinct roots of the equation and print all of them in ascending order.

Input

The first line contains three integer numbers A, B and C ($-10^5 \le A, B, C \le 10^5$). Any coefficient may be equal to 0.

Output

In case of infinite root count print the only integer -1. In case of no roots print the only integer 0. In other cases print the number of root on the first line and the roots on the following lines in the ascending order. Print roots with at least 5 digits after the decimal point.

Examples

input
1 -5 6
output
2 2.0000000000 3.0000000000

D. Fibonacci Sums

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fibonacci numbers have the following form:

$$F_1 = 1,$$
$$F_2 = 2,$$
$$F_i = F_{i-1} + F_{i-2}, i > 2.$$

Let's consider some non-empty set $S = \{s_1, s_2, ..., s_k\}$, consisting of **different** Fibonacci numbers. Let's find the sum of values of this set's elements:

$$\sum_{i=1}^{k} s_i = n$$

Let's call the set S a number n's *decomposition into Fibonacci sum*.

It's easy to see that several numbers have several decompositions into Fibonacci sum. For example, for 13 we have $13, 5+8, 2+3+8$ — three decompositions, and for 16: $3+13, 1+2+13, 3+5+8, 1+2+5+8$ — four decompositions.

By the given number n determine the number of its possible different decompositions into Fibonacci sum.

Input

The first line contains an integer t — the number of tests ($1 \le t \le 10^5$). Each of the following t lines contains one test.

Each test is an integer n ($1 \le n \le 10^{18}$).

Please do not use the `%lld` specificator to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specificator.

Output

For each input data test print a single number on a single line — the answer to the problem.

Examples

input
2 13 16
output
3 4

Note

Two decompositions are different if there exists a number that is contained in the first decomposition, but is not contained in the second one. Decompositions that differ only in the order of summands are considered equal.

B. Lucky Common Subsequence

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

In mathematics, a *subsequence* is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements. For example, the sequence `BDF` is a subsequence of `ABCDEF`. A *substring* of a string is a continuous subsequence of the string. For example, `BCD` is a substring of `ABCDEF`.

You are given two strings s_1, s_2 and another string called $virus$. Your task is to find the longest common subsequence of s_1 and s_2, such that it doesn't contain $virus$ as a substring.

Input

The input contains three strings in three separate lines: s_1, s_2 and $virus$ ($1 \leq |s_1|, |s_2|, |virus| \leq 100$). Each string consists only of uppercase English letters.

Output

Output the longest common subsequence of s_1 and s_2 without $virus$ as a substring. If there are multiple answers, any of them will be accepted.

If there is no valid common subsequence, output 0.

Examples

input
AJKEQSLOBSROFGZ OVGURWZLWVLUXTH OZ
output
ORZ

input
AA A A
output
0

A. Book Reading

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

Recently Luba bought a very interesting book. She knows that it will take t seconds to read the book. Luba wants to finish reading as fast as she can.

But she has some work to do in each of n next days. The number of seconds that Luba has to spend working during i-th day is a_i. If some free time remains, she can spend it on reading.

Help Luba to determine the minimum number of day when she finishes reading.

It is guaranteed that the answer doesn't exceed n.

Remember that there are 86400 seconds in a day.

Input

The first line contains two integers n and t ($1 \le n \le 100$, $1 \le t \le 10^6$) — the number of days and the time required to read the book.

The second line contains n integers a_i ($0 \le a_i \le 86400$) — the time Luba has to spend on her work during i-th day.

Output

Print the minimum day Luba can finish reading the book.

It is guaranteed that answer doesn't exceed n.

Examples

input
2 2 86400 86398
output
2

input
2 86400 0 86400
output
1

E. Desk Disorder

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

A new set of desks just arrived, and it's about time! Things were getting quite cramped in the office. You've been put in charge of creating a new seating chart for the engineers. The desks are numbered, and you sent out a survey to the engineering team asking each engineer the number of the desk they currently sit at, and the number of the desk they would like to sit at (which may be the same as their current desk). Each engineer must either remain where they sit, or move to the desired seat they indicated in the survey. No two engineers currently sit at the same desk, nor may any two engineers sit at the same desk in the new seating arrangement.

How many seating arrangements can you create that meet the specified requirements? The answer may be very large, so compute it modulo $1000000007 = 10^9 + 7$.

Input

Input will begin with a line containing N ($1 \leq N \leq 100000$), the number of engineers.

N lines follow, each containing exactly two integers. The i-th line contains the number of the current desk of the i-th engineer and the number of the desk the i-th engineer wants to move to. Desks are numbered from 1 to $2 \cdot N$. It is guaranteed that no two engineers sit at the same desk.

Output

Print the number of possible assignments, modulo $1000000007 = 10^9 + 7$.

Examples

input
4 1 5 5 2 3 7 7 3

output
6

input
5 1 10 2 10 3 10 4 10 5 5

output
5

Note

C. Anagram

time limit per test: 1 second
memory limit per test: 256 megabytes
input: input.txt
output: output.txt

String x is an *anagram* of string y, if we can rearrange the letters in string x and get exact string y. For example, strings "DOG" and "GOD" are anagrams, so are strings "BABA" and "AABB", but strings "ABBAC" and "CAABA" are not.

You are given two strings s and t of the same length, consisting of uppercase English letters. You need to get the anagram of string t from string s. You are permitted to perform the replacing operation: every operation is replacing some character from the string s by any other character. Get the anagram of string t in the least number of replacing operations. If you can get multiple anagrams of string t in the least number of operations, get the lexicographically minimal one.

The lexicographic order of strings is the familiar to us "dictionary" order. Formally, the string p of length n is lexicographically smaller than string q of the same length, if $p_1 = q_1, p_2 = q_2, \ldots, p_{k-1} = q_{k-1}, p_k < q_k$ for some k $(1 \le k \le n)$. Here characters in the strings are numbered from 1. The characters of the strings are compared in the alphabetic order.

Input

The input consists of two lines. The first line contains string s, the second line contains string t. The strings have the same length (from 1 to 10^5 characters) and consist of uppercase English letters.

Output

In the first line print z — the minimum number of replacement operations, needed to get an anagram of string t from string s. In the second line print the lexicographically minimum anagram that could be obtained in z operations.

Examples

input
ABA CBA
output
1 ABC

input
CDBABC ADCABD
output
2 ADBADC

Note

The second sample has eight anagrams of string t, that can be obtained from string s by replacing exactly two letters: "ADBADC", "ADDABC", "CDAABD", "CDBAAD", "CDBADA", "CDDABA", "DDAABC", "DDBAAC". These anagrams are listed in the lexicographical order. The lexicographically minimum anagram is "ADBADC".

A. Polo the Penguin and Strings

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

Little penguin Polo adores strings. But most of all he adores strings of length n.

One day he wanted to find a string that meets the following conditions:

1. The string consists of n lowercase English letters (that is, the string's length equals n), exactly k of these letters are distinct.
2. No two neighbouring letters of a string coincide; that is, if we represent a string as $s = s_1 s_2 \ldots s_n$, then the following inequality holds, $s_i \neq s_{i+1} (1 \leq i < n)$.
3. Among all strings that meet points 1 and 2, the required string is lexicographically smallest.

Help him find such string or state that such string doesn't exist.

String $x = x_1 x_2 \ldots x_p$ is *lexicographically less* than string $y = y_1 y_2 \ldots y_q$, if either $p < q$ and $x_1 = y_1, x_2 = y_2, \ldots, x_p = y_p$, or there is such number r $(r < p, r < q)$, that $x_1 = y_1, x_2 = y_2, \ldots, x_r = y_r$ and $x_{r+1} < y_{r+1}$. The characters of the strings are compared by their ASCII codes.

Input

A single line contains two positive integers n and k $(1 \leq n \leq 10^6, 1 \leq k \leq 26)$ — the string's length and the number of distinct letters.

Output

In a single line print the required string. If there isn't such string, print "-1" (without the quotes).

Examples

input
7 4
output
ababacd

input
4 7
output
-1

F. Buy One, Get One Free

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

A nearby pie shop is having a special sale. For each pie you pay full price for, you may select one pie of a strictly lesser value to get for free. Given the prices of all the pies you wish to acquire, determine the minimum total amount you must pay for all of the pies.

Input

Input will begin with an integer n ($1 \leq n \leq 500000$), the number of pies you wish to acquire. Following this is a line with n integers, each indicating the cost of a pie. All costs are positive integers not exceeding 10^9.

Output

Print the minimum cost to acquire all the pies.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
6 3 4 5 3 4 5
output
14

input
5 5 5 5 5 5
output
25

input
4 309999 6000 2080 2080
output
314159

Note

In the first test case you can pay for a pie with cost 5 and get a pie with cost 4 for free, then pay for a pie with cost 5 and get a pie with cost 3 for free, then pay for a pie with cost 4 and get a pie with cost 3 for free.

In the second test case you have to pay full price for every pie.

ICPC problems

http://codeforces.com/problemset/problem/96/A
http://codeforces.com/problemset/problem/876/A
http://codeforces.com/problemset/problem/873/B
http://codeforces.com/problemset/problem/883/B
http://codeforces.com/problemset/problem/877/C
http://codeforces.com/problemset/problem/877/D
http://codeforces.com/problemset/problem/873/D
http://codeforces.com/problemset/problem/863/E
http://codeforces.com/problemset/problem/845/F
http://codeforces.com/problemset/problem/883/H