Problem A
Allergen Testing
Time Limit: 1 Second, Memory Limit: 2G

You are in a chemistry class, and you are working with some compounds, exactly one of which you are allergic to. You have a fixed number of days to figure out which compound it is. You set up a number of sites on your arm for testing. On each day, you do the following exactly once:

1. Apply each compound to some (possibly empty) subset of sites on your arm. You can apply more than one compound to the same site.
2. Wait and see which sites demonstrate an allergic reaction.

A site demonstrates an allergic reaction if and only if the compound you are allergic to is applied to that site. If a site demonstrates an allergic reaction, it cannot be used on future days.

Compute the minimum number of sites you'll need on your arm to guarantee that you can determine exactly which compound you are allergic to within the given number of days.

Input

The first line of input contains a single integer t ($1 \leq t \leq 10^4$), which is the number of test cases that follow.

Each of the next t lines contains two integers n and d ($1 \leq n, d \leq 10^{18}$) describing a test case, where n is the number of compounds and d is the number of days.

Output

Output t lines. On each line output a single integer, which is the minimum number of sites on your arm necessary to discover which of the n compounds is the allergen within d days for that test case. Output the answers to the test cases in the order they appear in the input.

Sample Input 1	Sample Output 1
1 4 1	2

This page is intentionally left blank.

Problem B
A Tree and Two Edges
Time Limit: 3 Seconds, Memory Limit: 2G

Given a connected simple graph (with at most one edge between any pair of nodes) with n nodes and $n+1$ edges (that's a tree with two extra edges), answer a list of queries: for two distinct nodes, how many simple paths are there between them? A simple path is a path that does not repeat nodes.

Input

The first line of input contains two integers n ($4 \le n \le 5 \times 10^4$) and q ($1 \le q \le 5 \times 10^4$), where n is the number of nodes and q is the number of queries. The nodes are numbered from 1 to n.

Each of the next $n+1$ lines contains two integers a and b ($1 \le a < b \le n$) indicating that there is an edge in the graph between nodes a and b. All edges are distinct.

Each of the next q lines contains two integers u and v ($1 \le u < v \le n$). This is a query for the number of simple paths between nodes u and v.

Output

Output q lines. On each line output a single integer, which is the number of simple paths between the query nodes. Output the answers to the queries in the order they appear in the input.

Sample Input 1

```
4 6
1 2
1 3
1 4
2 3
2 4
1 2
1 3
1 4
2 3
2 4
3 4
```

Sample Output 1

```
3
3
3
3
3
4
```

Sample Input 2

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

Sample Output 2

```
2
2
4
1
```

Problem C
Broken Minimum Spanning Tree
Time Limit: 1 Second, Memory Limit: 2G

Ethan was tasked with finding a minimum spanning tree of a weighted, connected, undirected graph. However, he misunderstood the task and found a spanning tree that may not be minimal. To make his spanning tree a minimum spanning tree, you perform a sequence of edge swaps. An edge swap removes one edge from the spanning tree and adds an edge from the graph which is not currently in the spanning tree. After each edge swap, the tree must still be a spanning tree. What is the minimum number of edge swaps you must perform to fix Ethan's minimum spanning tree?

Input

The first line of input contains two integers n $(2 \leq n \leq 2{,}000)$ and m $(n-1 \leq m \leq 3{,}000)$, where n is the number of nodes in the graph and m is the number of edges in the graph. The nodes are numbered from 1 to n.

Each of the next m lines contains three integers u, v $(1 \leq u, v \leq n, u \neq v)$, and w $(1 \leq w \leq 10^9)$, signifying an edge connecting nodes u and v with weight w. The edges are numbered from 1 to m.

It is guaranteed that the graph is connected. The first $n - 1$ edges of the input are Ethan's initial spanning tree. The graph may not be simple; there can be multiple edges between the same pair of nodes.

Output

Output a single integer k, which is the minimum number of edge swaps needed to make the spanning tree a minimum spanning tree. Then output k lines, each with two integers a and b, where a is the number of the edge to remove and b is the number of the edge to add. If there are multiple sets of k edge swaps that work, any one will be accepted.

Sample Input 1	Sample Output 1

4 4
1 2 10
2 3 3
3 4 1
1 4 4
	1
	1 4

This page is intentionally left blank.

Problem D
Fail Fast
Time Limit: 4 Seconds, Memory Limit: 2G

A large software project has been written with many automated tests to help ensure quality. When a change is made to the source code, all of the automated tests are run in some sequence to help show that the code change has not broken some pre-existing functionality. The code change is accepted if and only if all automated tests pass.

Running all of these automated tests is costly, so as soon as there is one failure, the remaining tests are not run. Each test takes a certain amount of CPU time to perform. The cost of running a sequence of tests when there is a test failure is the sum of the CPU time used by the automated tests up to and including the failing test. The cost of running a sequence of tests where all tests pass is equal to 0.

A given automated test may rely on the output of another single automated test. For example, automated test **A** may depend on the output of automated test **B**. In such a case, test **B** must be executed before test **A**. Note that other tests may be run between test **B** and test **A**.

Based on an analysis of past data and an assessment of the code change, each automated test has a known probability of passing. The probability of any given test passing is independent of the probability of any other test passing.

Given the CPU time to execute each test, the probability of each test's failure, and the dependencies, determine an order in which the tests should be executed, sequentially, to minimize the expected cost of running the sequence of tests.

Input

The first line of input contains a single integer n ($1 \le n \le 10^5$), which is the number of automated tests.

Each of the next n lines contains three values, integer c ($1 \le c \le 10^6$), real number p ($0 < p < 1$), and integer d ($0 \le d \le n$), where c is the CPU time needed to run the automated test, p is the probability of the test passing, and d is the index of the test on which this one depends or 0 if the test has no dependency. The tests are indexed from 1 to n in the order in which they are given in the input.

Each probability is specified with at most six decimal digits. There are no cyclic dependencies.

Output

Output n lines, each line containing a single integer, giving the indices of the tests to run in sequence. The ordering must minimize the expected cost of running the tests to within 10^{-6} absolute or relative error of the expected cost of the optimal order. Any ordering which satisfies this constraint will be accepted.

Sample Input 1

```
4
100 0.5 0
200 0.1 1
10 0.5 2
10 0.9 0
```

Sample Output 1

```
4
1
2
3
```

Problem E
First Last
Time Limit: 1 Second, Memory Limit: 2G

Alice and Bob are playing a word game. They start with a list of words, and they alternate turns. Alice goes first; she chooses a starting word from the list. On each subsequent turn, the current player must choose a word from the list that starts with the same letter that ends the word chosen by the other player in the previous turn. No word can be used more than once. At some point one of them will not be able to choose a word; that player loses.

Assume Alice and Bob both play optimally. How many words from the list, when chosen as the first word by Alice, lead to a win for her?

Input

The first line of input contains a single integer n $(1 \leq n \leq 1{,}000)$ which is the number of words.

Each of the next n lines contains a single word consisting only of the lower-case letters '**a**' through '**z**'. Each word will be from two to fifteen letters long. All words will be distinct. There will be at most three distinct letters at the beginning and end of all words. Alice and Bob may only choose words from this list.

Output

Output a single integer, which is the number of words from the list which force a win for Alice if she chooses it first, and both Alice and Bob play optimally.

Sample Input 1	Sample Output 1

`3`
`attic`
`climb`
`alpha`
	`2`

Sample Input 2

```
22
agora
alpha
antic
aorta
apnea
arena
aroma
attic
basic
blurb
china
circa
civic
climb
cobra
cocoa
comic
comma
conic
crumb
cubic
cynic
```

Sample Output 2

```
6
```

Problem F
Four Square
Time Limit: 4 Seconds, Memory Limit: 2G

Inspired by a Piet Mondrian painting, you want to make a four-pane window out of four colored rectangles of glass. You are given the sizes of the four panes of glass. Can you arrange them into a square? You may rotate the panes. The panes should cover the entire square without overlap.

Input

The input consists of exactly four lines. Each of the four lines contains two integers w and h ($1 \leq w, h \leq 1{,}000$), representing the width and height of one of the four rectangular panes of glass.

Output

Output a single integer, which is 1 if the four panes can be arranged to form a square, and 0 otherwise.

Sample Input 1

```
1 1
1 1
1 1
1 1
```

Sample Output 1

```
1
```

Sample Input 2

```
3 1
3 3
2 2
3 3
```

Sample Output 2

```
0
```

This page is intentionally left blank.

Problem G
Frequent Flier
Time Limit: 2 Seconds, Memory Limit: 2G

An airline is offering a strange new rewards program to offer free flights to travelers.

The program can be parameterized with two integers m and k. Within any m consecutive months, a traveler must pay for at least k of those flights (if there are fewer than k flights in that interval, all of those flights must be paid for). Other flights within that interval are free. Note that this condition needs to be true for all m-month intervals, including all of the ones that start before your first flight.

You have a schedule of the number of flights you'll take over the next many months. You want to know the minimum number of flights you'll need to pay for.

Input

The first line of input contains three integers n, m ($1 \leq n, m \leq 2 \times 10^5$) and k ($1 \leq k \leq 10^9$), where n is the number of consecutive months ahead that you have flights planned, and m and k are the parameters of the airline's rewards program.

Each of the next n lines contains an integer f ($1 \leq f \leq 10^9$), which is the number of flights you plan to take during that month.

Output

Output a single integer, which is the minimum number of planned flights that you must pay for.

Sample Input 1	Sample Output 1

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

This page is intentionally left blank.

Problem H
Game Show Elimination
Time Limit: 8 Seconds, Memory Limit: 2G

You are running an elimination-style game show, where players are eliminated one at a time until only one remains. Based on what you know about the contestants, you are trying to predict the results.

Each week, the remaining contestants take part in a competition, where each contestant's score is based on their skill level. Because this is a silly game show, a contestant's score is a random number that falls within their unique skill range. The contestants are then ranked from highest to lowest based on their scores. (Since scores are all real numbers, there is zero probability that there is a tie.)

The winner of this week's competition chooses who is eliminated that week. However, all players believe they should pick whoever did the best after them in the week's competition, so they always choose to eliminate the second place contestant. That contestant is eliminated and goes home. The show continues, week after week, until there is only one contestant left.

The contestants' final ranks are based on when they left the competition. The last contestant to leave (the winner!) is assigned rank 1, the second-to-last gets rank 2, and so on until the first person to leave is assigned the lowest rank.

Given information about the contestants' skill levels, compute the expected ranks of each player.

Input

The single line of input contains two integers n ($2 \leq n \leq 1{,}000$) and k ($2 \leq k \leq 10$), where n is the number of game show contestants, and k determines the skill range of the contestants.

Contestants are numbered from 1 to n. The skill range of contestant i is from i to $i + k$ inclusive, and their score each week is a randomly assigned real number in this range.

Output

Output n lines. Each line contains a single real number, which is the expected rank of a contestant. The expected ranks must be listed in contestant number order. The answers are accepted within absolute or relative error of at most 10^{-6}.

Sample Input 1

```
3 2
```

Sample Output 1

```
2.109375
2.625000
1.265625
```

Problem I
Power of Divisors
Time Limit: 1 Second, Memory Limit: 2G

Consider a positive integer n. Let $f(n)$ be the number of positive integer divisors of n. For example, if $n = 8$ then $f(n) = 4$, since the divisors of 8 are 1, 2, 4 and 8.

Now, consider a positive integer x. What is the smallest value of n such that $n^{f(n)} = x$?

Input

The single line of input contains a single integer x ($1 \leq x \leq 10^{18}$). This is the x of the statement above.

Output

Output a single integer, which is the smallest value of n such that $n^{f(n)} = x$, or -1 if no such value of n exists.

Sample Input 1	Sample Output 1
15625	25

Sample Input 2	Sample Output 2
64000000	20

Sample Input 3	Sample Output 3
65536	-1

This page is intentionally left blank.

Problem J
Repetitive String Invention
Time Limit: 2 Seconds, Memory Limit: 2G

Lulu has a string consisting of lowercase English letters. She would like to make her string *Repetitive*. A repetitive string has an even number of characters, and the first half of the string exactly matches the second half of the string. For example, "**lulu**", "**abcabc**" and "**xx**" are repetitive strings, while "**xyx**" and "**abac**" are not.

To get a repetitive string, Lulu can take two non-overlapping, non-empty substrings from her string and concatenate them together. The substrings must be concatenated in the order that they appear in her string.

She's wondering, what is the number of ways she can choose two substrings to make a repetitive string? Two ways are different if at least one of the substrings Lulu uses comes from a different part of her string.

Consider the string "**aaaa**".

- There are six ways for Lulu to form the repetitive string "**aa**": by matching each "**a**" with each subsequent "**a**" (1+2, 1+3, 1+4, 2+3, 2+4, 3+4).
- There are also three ways for her to form "**aaaa**": "**a**"+"**aaa**", "**aa**"+"**aa**" and "**aaa**"+"**a**".

So there are nine ways for Lulu to form a repetitive string by concatenating non-overlapping, non-empty substrings of "**aaaa**" in order.

Input

The single line of input contains a single string s ($1 \le |s| \le 800, s \in \{\mathbf{a} - \mathbf{z}\}^*$). This is Lulu's string.

Output

Output a single integer, which is the number of ways Lulu can concatenate two non-overlapping, non-empty substrings from her string in order to get a repetitive string.

Sample Input 1

```
aaaa
```

Sample Output 1

```
9
```

Sample Input 2

```
axabxbcxcdxd
```

Sample Output 2

```
22
```

Problem K
Space Alignment
Time Limit: 1 Second, Memory Limit: 2G

You are collaborating with a few other programmers on a coding project. To your horror, you discover that some people have been using tabs to indent code while other people have been using spaces to indent code within the same file. Apparently these people didn't notice that anything was amiss when they did so.

You wish to replace the tabs with spaces so that the file has consistent indenting. Consistent indenting means that for every line at a nesting depth of k, $k \geq 0$, the first non-whitespace character in the line should be preceded by exactly $k \cdot i$ spaces, where $i > 0$.

Is it possible to replace each tabs with a fixed number of spaces to repair the file and have consistent indenting?

Input

The first line of input contains a single integer n ($2 \leq n \leq 100$), which is the number of lines of code.

Each of the next n lines contains a string, consisting of a sequence of the characters '**s**' (representing a space) and/or '**t**' (representing a tab), followed by a single '**{**' or '**}**'. Each line will have at most $1{,}000$ characters.

The first line is guaranteed to be a single '**{**', the last line is guaranteed to be a single '**}**', and the braces throughout the data are guaranteed to match; that is, looking at only those characters as a single string, it is always possible to repeatedly remove a substring "**{}**" until you are left with the empty string.

Output

Output a single integer, which is the smallest number of spaces greater than zero to replace each tab with and achieve consistent indentation, or -1 if it isn't possible.

Sample Input 1

```
10
{
ss{
sts{
tt}
t}
t{
ss}
}
{
}
```

Sample Output 1

```
2
```

Problem L
Splitting Pairs
Time Limit: 1 Second, Memory Limit: 2G

Alice and Bob are playing a modified game of Nim. Initially, there are some non-empty piles of stones in front of them. They take turns, and Alice takes the first turn.

On a single turn, a player must do the following actions in order:

- Remove some number of piles of stones — at least one but no more than half the number of piles.
- Choose the same number of piles of remaining stones, and split each of those piles into two non-empty piles.

Notice that after each valid move, there should be the same number of non-empty piles of stones as at the start of the game. A player who cannot perform all the actions on their turn loses the game.

You are given many games, and for each one, you'd like to determine who would win if both players play optimally.

Input

The first line of input contains an integer t ($1 \leq t \leq 1{,}000$), which is the number of games Alice and Bob play.

Each game is represented on two lines. The first line of each game contains an integer n ($2 \leq n \leq 50$), which is the number of piles of stones.

The next line of the game contains n space-separated integers s ($1 \leq s \leq 10^{12}$), which are the number of stones in each pile.

Output

Output t lines. For each game, output a single line with a single integer, which is 1 if Alice wins and 0 if Bob wins. Assume Alice takes the first turn, and both players play optimally. Output the game results in the order the games appear in the input.

Sample Input 1

```
4
3
1 1 1
3
1 1 2
3
2 2 2
4
4 4 4 4
```

Sample Output 1

```
0
1
0
1
```

Problem M
Who Watches the Watchmen?
Time Limit: 4 Seconds, Memory Limit: 2G

There are some sentry drones guarding a top-secret facility. Each sentry is stationary at some point in 3D space, and faces in some viewing direction.

With recent advances in artificial intelligence, the owners of the facility have come to the realization that the greatest threats to the facility are not intruders, but the sentries themselves! For security, they want to adjust the sentries such that every sentry is watching another sentry and every sentry is seen by exactly one other sentry.

It costs 1 unit of energy to change the viewing direction of a sentry and $1{,}000$ units of energy to move a sentry to a new location. Note that these operations are independent. It costs $1{,}001$ units of energy in total to change both a sentry's position and viewing direction. No two sentries can ever be at the same position at the end of a move. After being moved, a sentry's position might not be on an integral lattice point.

A sentry at location (x, y, z) facing direction (vx, vy, vz) can see any point $(x+t\cdot vx, y+t\cdot vy, z+t\cdot vz)$ for $t \geq 0$ so long as there is no other sentry directly between it and the point. What is the minimum amount of energy needed to reposition the sentries so that each sentry can be seen by exactly one other sentry?

Input

The first line of input contains a single integer n ($1 \leq n \leq 500$), which is the number of sentries.

Each of the next n lines contains six integers x, y, z, vx, vy and vz, indicating that there is a sentry at (x, y, z) facing in direction (vx, vy, vz). All values will be in the range from -10^6 to 10^6, inclusive. At least one of vx, vy or vz will be non-zero. No two sentries will be at the same position.

Output

Output a single integer, which is the minimum amount of energy needed to reposition the sentries so that each sentry can be seen by exactly one other sentry, or -1 if it isn't possible.

Sample Input 1

```
4
66 45 10 73 39 36
95 14 26 47 84 59
14 66 89 89 36 78
16 27 94 79 24 24
```

Sample Output 1

```
4
```