Problem A. Bonus Project

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

There is a team of n software engineers numbered from 1 to n. Their boss promises to give them a bonus if they complete an additional project. The project requires k units of work in total. The bonus promised to the i-th engineer is a_i burles. The boss doesn't assign specific tasks to engineers; it is expected that every engineer will voluntarily complete some integer amount of work units. The bonus will be paid to the entire team only if the project is completed; in other words, if the total amount of voluntary work units on the project is greater than or equal to k.

The amount of work that can be performed by each engineer is not limited. However, all engineers value their labour. The i-th engineer estimates one unit of their work as b_i burles. If the bonus is paid, the benefit s_i of the i-th engineer for completing c units of work is defined as $s_i = a_i - c \cdot b_i$. If the bonus is not paid, the engineer will not volunteer to do any work.

Engineers work together for many years, so they know how the bonus is going to be distributed and how much their colleagues value the labour. That is, all a_i and all b_i are known to every engineer in the team.

Engineers are eager to get the bonus, so they agreed on the following process of work distribution between them:

- the first engineer says: "I will complete c_1 units of work", where c_1 is a non-negative integer;
- then, the second engineer says: "I will complete c_2 units of work", where c_2 is a non-negative integer;
- ...and so on;
- finally, the n-th engineer says: "I will complete c_n units of work", where c_n is a non-negative integer.

Every engineer voices c_i in a way to maximize their own benefit s_i. If the expected benefit is going to be zero, an engineer will still agree to work to get the experience and to help their colleagues obtain the bonus. However, if the benefit is expected to be negative for some reason (an engineer needs to perform an excessive amount of work or the project is not going to be completed), that engineer will not work at all (completes zero amount of work units).

Given that every engineer acts perfectly, your task is to find out the numbers c_i voiced by every engineer.

Input

The first line contains two integers n and k ($1 \le n \le 1000$; $1 \le k \le 10^6$) — the number of engineers in the company and the number of work units the project requires, respectively.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$), where a_i is the bonus which will be paid to the i-th engineer if the project is completed.

The third line contains n integers $b_1, b_2, \dots, b_n$ ($1 \le b_i \le 1000$), where b_i is the work unit cost for the i-th engineer.

Output

Print n integers $c_1, c_2, \dots, c_n$ ($0 \le c_i \le k$) — the amount of work completed by each engineer given that every engineer behaves optimally. Note that the answer is unique.

Examples

standard input	standard output
3 6 4 7 6 1 2 3	1 3 2
3 12 4 7 6 1 2 3	0 0 0
3 11 6 7 8 1 2 3	6 3 2

Note

In the first example, engineers distributed the work across them and got the bonus, even though the benefit for the third engineer is zero.

In the second example, the bonus project requires too many work units to complete, so it's more beneficial for engineers not to work at all.

Problem B. Make It Equal

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

You are given an integer array a of size n. The elements of the array are numbered from 1 to n.

You can perform the following operation any number of times (possibly, zero): choose an index i from 1 to n; decrease a_i by 2 and increase $a_{(i \bmod n)+1}$ by 1.

After you perform the operations, all elements of the array should be **non-negative equal integers**.

Your task is to calculate the minimum number of operations you have to perform.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains a single integer n ($2 \le n \le 2 \cdot 10^5$).

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$).

Additional constraint on the input: the sum of n over all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer — the minimum number of operations you have to perform. If it is impossible to make all elements of the array equal, print `-1`.

Example

standard input	standard output
3 2 1 1 3 1 3 2 4 2 1 2 6	0 -1 3

Problem C. DIY

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

You are given a list of n integers $a_1, a_2, \ldots, a_n$. You need to pick 8 elements from the list and use them as coordinates of four points. These four points should be corners of a rectangle which has its sides parallel to the coordinate axes. Your task is to pick coordinates in such a way that the resulting rectangle has the maximum possible area. The rectangle can be degenerate, i. e. its area can be 0. Each integer can be used as many times as it occurs in the list (or less).

Input

The first line contains one integer t ($1 \le t \le 25\,000$) — the number of test cases.

The first line of each test case contains one integer n ($8 \le n \le 2 \cdot 10^5$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$).

Additional constraint on the input: the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print the answer as follows:

- if it is impossible to construct a rectangle which meets the constraints from the statement, print a single line containing the word `NO` (case-insensitive);
- otherwise, in the first line, print `YES` (case-insensitive). In the second line, print 8 integers $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ — the coordinates of the corners of the rectangle. You can print the corners in any order.

Example

standard input	standard output
3 16 -5 1 1 2 2 3 3 4 4 5 5 6 6 7 7 10 8 0 0 -1 2 2 1 1 3 8 0 0 0 0 0 5 0 5	YES 1 2 1 7 6 2 6 7 NO YES 0 0 0 5 0 0 0 5

Problem D. Divide OR Conquer

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

You are given an array $[a_1, a_2, \ldots a_n]$ consisting of integers between 0 and 10^9. You have to split this array into several segments (possibly one) in such a way that each element belongs to exactly one segment.

Let the first segment be the array $[a_{l_1}, a_{l_1+1}, \ldots, a_{r_1}]$, the second segment be $[a_{l_2}, a_{l_2+1}, \ldots, a_{r_2}]$, ..., the last segment be $[a_{l_k}, a_{l_k+1}, \ldots, a_{r_k}]$. Since every element should belong to exactly one array, $l_1 = 1$, $r_k = n$, and $r_i + 1 = l_{i+1}$ for each i from 1 to $k-1$. The split should meet the following condition: $f([a_{l_1}, a_{l_1+1}, \ldots, a_{r_1}]) \le f([a_{l_2}, a_{l_2+1}, \ldots, a_{r_2}]) \le \cdots \le f([a_{l_k}, a_{l_k+1}, \ldots, a_{r_k}])$, where $f(a)$ is the bitwise OR of all elements of the array a.

Calculate the number of ways to split the array, and print it modulo $998\,244\,353$. Two ways are considered different if the sequences $[l_1, r_1, l_2, r_2, \ldots, l_k, r_k]$ denoting the splits are different.

Input

The first line contains an integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the array a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$) — the elements of the given array.

Output

Print one integer — the number of ways to split the array, taken modulo $998\,244\,353$.

Examples

standard input	standard output
3 1 2 3	4
5 1000 1000 1000 1000 1000	16
3 3 4 6	3

Note

In the first two examples, every way to split the array is valid.

In the third example, there are three valid ways to split the array:

- $k = 3$; $l_1 = 1, r_1 = 1, l_2 = 2, r_2 = 2, l_3 = 3, r_3 = 3$; the resulting arrays are $[3]$, $[4]$, $[6]$, and $3 \le 4 \le 6$;
- $k = 2$; $l_1 = 1, r_1 = 1, l_2 = 2, r_2 = 3$; the resulting arrays are $[3]$ and $[4, 6]$, and $3 \le 6$;
- $k = 1$; $l_1 = 1, r_1 = 3$; there will be only one array: $[3, 4, 6]$.

If you split the array into two arrays $[3, 4]$ and $[6]$, the bitwise OR of the first array is 7, and the bitwise OR of the second array is 6; $7 > 6$, so this way to split the array is invalid.

Problem E. Barrels

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Suppose you have n water barrels standing in a row, numbered from 1 to n.

All barrels are equal and have a bottom area equal to one unit, so the volume of the water inside a barrel is equal to the height of the water column. Initially, the i-th barrel has v_i units of water.

Adjacent barrels are connected by pipes. In other words, for each i from 1 to $n-1$, barrels i and $i+1$ are connected by a single horizontal pipe at height h_i. The widths of the pipes are negligible. These pipes allow water to flow between barrels.

Now you want to play with barrels. Your plan is to maximize the volume of the water in the first barrel by throwing clay into barrels. In one step, you can choose any barrel and throw one unit of clay into it. One unit of clay has the same volume as one unit of water. Clay is heavier than water and doesn't mix with it, so it falls to the bottom of the barrel, distributing evenly.

Clay has a sticky structure, so it seals pipes if the clay column is high enough. More formally, suppose the pipe is at height h. **If the height of the clay column is also h (or lower), the pipe is working**. But the moment you add more clay into the barrel, the pipe becomes sealed instantly, preventing any water from moving between barrels.

You have a mountain of clay, so you can repeat the step described above any number of times. However, between the steps, you have to wait until the water reaches the new equilibrium.

What is the maximum water volume you can collect in the first barrel?

Assume that the barrels are high enough, so the water doesn't overflow, and the pipe widths are negligible.

Input

The first line contains a single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of barrels.

The second line contains n integers $v_1, v_2, \dots, v_n$ ($0 \le v_i \le 10^6$), where v_i is the initial water volume in the i-th barrel.

The third line contains $n-1$ integers $h_1, h_2, \dots, h_{n-1}$ ($1 \le h_i \le 10^6$), where h_i is the height of the pipe between the i-th and the $(i+1)$-st barrel.

Additional constraint on the input: the given water heights are in an equilibrium.

Output

Print a single number — the maximum volume of water in the first barrel. Your answer is considered correct if its absolute or relative error does not exceed 10^{-6}.

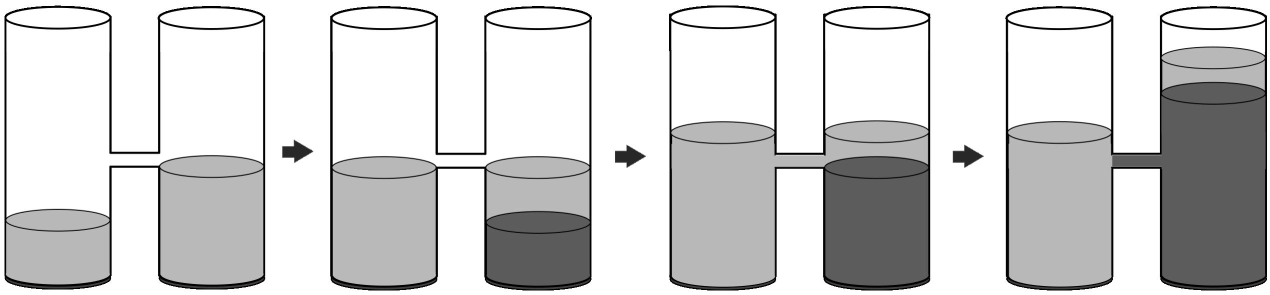
Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Examples

standard input	standard output
2 1 2 2	2.500000000000000
3 3 0 0 6 9	3.000000000000000
5 10 0 0 0 5 11 1 2 5	11.916666666666667

Note

An optimal strategy for the first example is shown in the picture below:



Problem F. Alternative Platforms

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Suppose you are working in the Ministry of Digital Development of Berland, and your task is to monitor the industry of video blogging.

There are n bloggers in Berland. Recently, due to the poor state of the main video platform in Berland, two alternative platforms were introduced. That's why bloggers started to reupload their videos to these alternative platforms. You've got the statistics that the i-th blogger uploaded v_i videos to the first alternative platform and r_i videos to the second alternative platform.

You think that a potential user will be upset if even at least one of his favorite bloggers doesn't upload anything. However, if a blogger uploads videos to both platforms, the user will watch that blogger on the platform where more videos are available. So, you've come up with the following function to estimate user experience. Suppose a user watches k bloggers $b_1, b_2, \ldots, b_k$; then, let user experience be

$$E(b_1, \ldots, b_k) = \max\left(\min_{i=1..k} v[b_i], \min_{i=1..k} r[b_i]\right).$$

In order to get some statistics, you want to calculate the value avg_k that is equal to an average experience among all subsets of bloggers of size k. Also, you have to calculate avg_k for each k from 1 to n.

Since answers may be too large, print them modulo $998\,244\,353$.

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of bloggers.

The second line contains n integers $v_1, v_2, \ldots, v_n$ ($0 \le v_i \le 10^6$), where v_i is the number of videos of the i-th blogger on the first alternative platform.

The third line contains n integers $r_1, r_2, \ldots, r_n$ ($0 \le r_i \le 10^6$), where r_i is the number of videos of the i-th blogger on the second alternative platform.

Output

Print n integers $avg_1, avg_2, \ldots, avg_n$.

It can be proven that avg_k may be represented as an irreducible fraction $\frac{x}{y}$ where $y \not\equiv 0 \pmod{998\,244\,353}$. So, print avg_k in a form $x \cdot y^{-1} \bmod 998\,244\,353$.

Examples

standard input	standard output
3 2 1 2 1 2 1	2 332748119 1
4 5 5 5 5 0 0 0 0	5 5 5 5
5 1 9 3 7 5 2 4 6 8 5	6 4 3 199648873 2

Note

In the first example, 332748119 is $\frac{4}{3}$. In the third example, 199648873 is $\frac{12}{5}$.

Problem G. Guess One Character

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

This is an interactive problem. You have to use `flush` operation right after printing each line. For example, in C++ you should use the function `fflush(stdout)` or `cout.flush()`, in Java or Kotlin — `System.out.flush()`, and in Python — `sys.stdout.flush()`.

The jury has a string s consisting of characters `0` and/or `1`. The length of this string is n.

You can ask the following queries:

- $1\ t$ — "how many times does t appear in s as a contiguous substring?" Here, t should be a string consisting of characters `0` and/or `1`; its length should be at least 1 and at most n. For example, if the string s is `111011` and the string t is `11`, the response to the query is 3.

You have to guess at least one character in the string s by asking no more than 3 queries. Note that giving the answer does not count as a query.

In every test and in every test case, the string s is fixed beforehand.

Interaction Protocol

Initially, the jury program sends one integer t ($1 \le t \le 1000$) — the number of test cases.

At the start of each test case, the jury program sends one integer n ($2 \le n \le 50$) — the length of the string.

After that, your program can submit queries to the jury program by printing the following line **(do not forget to flush the output after printing a line!)**:

- $1\ t$ means asking a query "how many times does t appear in s as a contiguous substring?"

For every query, the jury prints one integer on a separate line. It is either:

- the answer to your query, if the query is correct, and you haven't exceeded the query limit;
- or the integer -1, if your query is incorrect (for example, the constraint $1 \le |t| \le n$ is not met or the string t contains characters other than `0` and `1`) or if you have asked too many queries while processing the current test case.

To submit the answer, your program should send a line in the following format **(do not forget to flush the output after printing a line!)**:

- $0\ i\ c$, where $1 \le i \le n$ and c is either `0` or `1`, meaning that $s_i = c$.

If your guess is correct, the jury program will print one integer 1 on a separate line, indicating that you may proceed to the next test case (or terminate the program, if it was the last test case) and that the number of queries you have asked is reset. If it is not correct, the jury program will print one integer -1 on a separate line.

After your program receives -1 as the response, it should immediately terminate. This will lead to your submission receiving the verdict "Wrong Answer". If your program does not terminate, the verdict of your submission is undefined.

Example

standard input	standard output
`3     // 3 test cases`	
`3     // the length of the string is 3`	
	`1 101 // how many times 101 occurs`
`1     // 101 occurs once`	
	`0 2 0 // guess: s[2] is 0`
`1     // guessed correctly`	
`2     // the length of the string is 2`	
	`1 00  // how many times 00 occurs`
`0     // 00 occurs zero times`	
	`1 0   // how many times 0 occurs`
`0     // 0 occurs zero times`	
	`0 1 1 // guess: s[1] is 1`
`1     // guessed correctly`	
`2     // the length of the string is 2`	
	`1 1   // how many times 1 occurs`
`1     // 1 occurs once`	
	`1 01  // how many times 01 occurs`
`0     // 01 occurs zero times`	
	`0 2 0 // guess: s[2] is 0`
`1     // guessed correctly`	

Note

In the example, there are 3 test cases: `101`, `11` and `10`. Note that everything after the `//` sign is a comment that explains which line means what in the interaction. **The jury program won't print these comments in the actual problem, and you shouldn't print them**. The empty lines are also added for your convenience, **the jury program won't print them, and your solution should not print any empty lines**.

Problem H. Galactic Council

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

Monocarp plays a computer game. In this game, he maintains a space empire. The empire is governed by n political parties. Initially, every party has political power equal to 0, and there is no ruling party.

During each of the next m turns, the following happens:

1. initially, Monocarp **has to** choose which party he supports. He can support any party, **except for the ruling party**. When Monocarp supports a party, its political power is increased by 1. If Monocarp supports the i-th party during the j-th turn, his score increases by $a_{i,j}$ points;
2. then, the elections happen, and the party with the maximum political power is chosen as the ruling party (if there are multiple such parties, **the party with the lowest index among them is chosen**). The former ruling party is replaced, unless it is chosen again;
3. finally, an event happens. At the end of the j-th turn, the party p_j must be the ruling party to prevent a bad outcome of the event, otherwise Monocarp loses the game.

Determine which party Monocarp has to support during each turn so that he doesn't lose the game due to the events, and the score he achieves is the maximum possible. Initially, Monocarp's score is 0.

Input

The first line contains two integers n and m ($2 \le n, m \le 50$) — the number of political parties and the number of turns, respectively.

The second line contains m integers $p_1, p_2, \dots, p_m$ ($1 \le p_j \le n$), where p_j is the index of the party which should be the ruling party at the end of the j-th turn.

Then n lines follow. The i-th of them contains m integers $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ($1 \le a_{i,j} \le 10^5$), where $a_{i,j}$ is the amount of points Monocarp gets if he supports the i-th party during the j-th turn.

Output

If Monocarp loses the game no matter how he acts, print one integer -1.

Otherwise, print m integers $c_1, c_2, \dots, c_m$ ($1 \le c_j \le m$), where c_j is the index of the party Monocarp should support during the j-th turn. If there are multiple answers, print any of them.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>2 3
2 1 2
1 2 3
4 5 6</td><td>2 1 2</td></tr>
<tr><td>3 5
1 1 1 2 1
1 1 1 1 1
10 5 7 8 15
7 10 9 8 15</td><td>1 3 2 2 1</td></tr>
<tr><td>3 5
1 1 1 1 1
1 1 1 1 1
10 5 7 8 15
7 10 9 8 15</td><td>-1</td></tr>
</table>

Problem I. Polyathlon

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

Berland is this year's host country of the International Collegiate Polyathlon Competition! Similar to biathlon being a competition of two sports, polyathlon is a competition of many sports. This year, there are m sports. Also, there are n participants in the event. The sports are numbered from 1 to m, and the participants are numbered from 1 to n.

Some participants are skilled in multiple sports. You are given a binary matrix $n \times m$ such that the j-th character of the i-th row is `1` if the i-th participant is skilled in the j-th sport, and `0`, otherwise. It's also known that, for each pair of participants, there exists at least one sport such that one of them is skilled in that sport and the other one isn't.

The order of sports in the competition is determined at the opening ceremony. Historically, it's done by the almighty Random Number Generator. A random number x from 1 to m is rolled. The competition then starts with the sport x, then the sport $(x \bmod m + 1)$ is played, then the sport $((x + 1) \bmod m + 1)$, and so on.

Each sport is played as follows. If all remaining participants (all participants which are not eliminated yet) are not skilled in that sport, everyone goes through to the next sport. Otherwise, all skilled participants go through to the next sport, and all unskilled participants are eliminated from the competition. Once there is a single participant remaining in the competition, the competition ends, and that participant is declared the winner.

As an organizer of the competition, you are curious of the possible outcomes of the competition beforehand (not that you are going to rig the random roll, how could you possibly think that...). For each sport x, print the index of the winner if the competition starts with the sport x.

Input

The first line contains two integers n and m ($2 \le n, m \le 10^6$; $n \le 2^m$; $nm \le 2 \cdot 10^6$) — the number of participants and the number of sports, respectively.

The i-th of the next n lines contains a binary string consisting of exactly m characters `0` or `1` — the skillset of the i-th participant. If the j-th character is `1`, the i-th participant is skilled in the j-th sport. If it's `0`, the i-th participant is not skilled in the j-th sport.

Additional constraint on the input: for each pair of participants, there exists at least one sport such that one of them is skilled in that sport and the other one isn't. In other words, all n binary strings are pairwise distinct.

Output

Print m integers. For each x from 1 to m, print the index of the winner if the competition starts with the sport x.

Example

standard input	standard output
3 5 10010 01100 10101	3 2 3 1 3

Problem J. Waiting for...

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Monocarp is waiting for a bus at the bus stop. Unfortunately, there are many people who want to ride a bus too.

You are given a list of events of two types:

- `B` b_i — a bus with b_i free seats arrives at the stop;
- `P` p_i — p_i people arrive at the stop.

These events are listed in a chronological order.

When a bus arrives, the following happens. All people at the bus stop (except for Monocarp) try to enter the bus. If there are enough free seats for all of them, then they all enter the bus. Otherwise, some people remain at the bus stop (the number of people who enter the bus is equal to the number of free seats).

If there is still at least one free seat after all people (except for Monocarp) enter the bus, then Monocarp can decide to enter this bus as well (but he might choose to wait for another bus). For each bus, you have to determine if it is possible for Monocarp to take that bus.

Input

The first line contains one integer n ($1 \le n \le 10^3$) — the number of events.

Then, n lines follow. The i-th of them contains the description of the i-th event in one of the two following formats:

- `B` b_i ($1 \le b_i \le 10^6$) — a bus with b_i free seats arrives at the stop;
- `P` p_i ($1 \le p_i \le 10^6$) — p_i people arrive at the stop.

Additional constraint on the input: there is at least one event of type `B`.

Output

For each event of type `B`, print `YES` if it is possible for Monocarp to take the corresponding bus, or `NO` otherwise (case-insensitive).

Example

standard input	standard output
10 P 2 P 5 B 8 P 14 B 5 B 9 B 3 P 2 B 1 B 2	YES NO NO YES NO YES

Problem K. Grid Walk

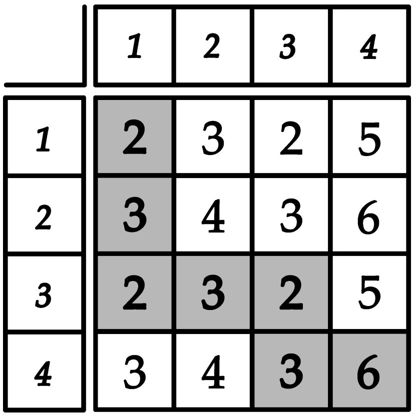
Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

You have an $n \times n$ grid and two integers a and b. Both the rows and the columns are numbered from 1 to n. Let's denote the cell at the intersection of the i-th row and the j-th column as (i, j).

You are standing in the cell $(1, 1)$ and want to move into the cell (n, n).

Suppose you are in the cell (i, j); in one step, you can move either into the cell $(i, j + 1)$ or into the cell $(i + 1, j)$ if the corresponding cells exist.

Let's define the cost of the cell (i, j) as $c(i, j) = \gcd(i, a) + \gcd(j, b)$ (here, $\gcd(x, y)$ denotes the greatest common divisor of x and y). The cost of the route from $(1, 1)$ to (n, n) is the sum of costs of the visited cells (including the starting cell and the finishing cell).



Find the route with minimum possible cost and print its cost.

Input

The only line contains three integers n, a, and b ($2 \le n \le 10^6$; $1 \le a, b \le 10^6$).

Output

Print one integer — the cost of the cheapest route from $(1, 1)$ to (n, n).

Examples

standard input	standard output
`4 2 4`	`21`
`10 210 420`	`125`

Note

The first example is described in the picture above.

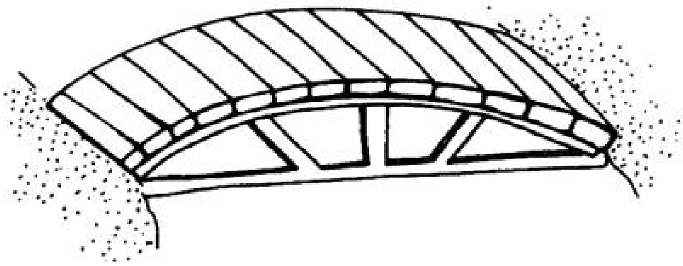
Problem L. Bridge Renovation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Recently, Monocarp started working as a director of a park located near his house. The park is quite large, so it even has a small river splitting it into several zones. Several bridges are built across this river. Three of these bridges are especially old and need to be repaired.

All three bridges have the same length but differ in width. Their widths are 18, 21 and 25 units, respectively.

During the park renovation process, Monocarp has to replace the old planks that served as the surface of the bridges with the new ones.



Planks are sold with a standard length of 60 units. Monocarp already knows that he needs n planks for each bridge. But since the bridges have different widths, he needs n planks of length 18 for the first bridge, n planks of length 21 for the second one, and n planks of length 25 for the last one.

Workers in charge of renovation have no problem with cutting planks into parts but refuse to join planks, since it creates weak spots and looks ugly.

Monocarp wants to buy as few planks as possible but struggles to calculate the required number of planks. Can you help him?

Input

The first and only line contains a single integer n ($1 \le n \le 1000$) — the number of planks required for each of the three bridges.

Output

Print a single integer — the minimum number of planks of standard length (60 units) Monocarp needs to cover all three bridges if the planks can be cut into parts.

Examples

standard input	standard output
1	2
3	4
1000	1167

Note

In the first example, it is possible to cut one plank of length 60 into three planks with lengths 25, 18 and 17, and cut another plank of length 60 into two planks with lengths 39 and 21. That way, Monocarp will have all the required planks.

Problem M. Royal Flush

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

Consider the following game. There is a deck, which consists of cards of n different suits. For each suit, there are 13 cards in the deck, all with different ranks (the ranks are 2, 3, 4, ..., 10, Jack, Queen, King and Ace).

Initially, the deck is shuffled randomly (all $(13n)!$ possible orders of cards have the same probability). You draw 5 topmost cards from the deck. Then, every turn of the game, the following events happen, in the given order:

1. if the cards in your hand form a *Royal Flush* (a 10, a Jack, a Queen, a King, and an Ace, all of the same suit), you win, and the game ends;
2. if you haven't won yet, and the deck is empty, you lose, and the game ends;
3. if the game hasn't ended yet, you may choose any cards from your hand (possibly, all of them) and discard them. When a card is discarded, it is removed from the game;
4. finally, you draw cards from the deck, until you have 5 cards or the deck becomes empty.

Your goal is to find a strategy that allows you to win in the minimum expected number of turns. Note that the turn when the game ends is not counted (for example, if the 5 cards you draw initially already form a *Royal Flush*, you win in 0 turns).

Calculate the minimum possible expected number of turns required to win the game.

Input

The only line contains one integer n ($1 \le n \le 4$) — the number of suits used in the game.

Output

Print the minimum expected number of turns.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}. Formally, let your answer be a, and the jury's answer be b. Your answer will be accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Examples

standard input	standard output
1	3.598290598
2	8.067171309

Problem N. Fixing the Expression

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

An *expression* is a string consisting of three characters, where the first and the last characters are digits (from 0 to 9), and the middle character is a comparison symbol (`<`, `=` or `>`).

An expression is *true* if the comparison symbol matches the digits (for example, if the first digit is strictly less than the last digit, the comparison symbol should be `<`).

For example, the expressions `1<3`, `4>2`, `0=0` are true, while `5>5`, `7<3` are not.

You are given a string s, which is an expression. Change as few characters as possible so that s becomes a true expression. Note that if s is already true, you should leave it as it is.

Input

The first line contains one integer t ($1 \le t \le 300$) — the number of test cases.

Each test case consists of one line containing the string s ($|s| = 3$, the first and the last characters of s are digits, the second character is a comparison symbol).

Output

For each test case, print a string consisting of 3 characters — a true expression which can be obtained by changing as few characters as possible in s. If there are multiple answers, print any of them.

Example

standard input	standard output
5 3<7 3>7 8=9 0=0 5<3	3<7 8>7 8<9 0=0 0<3