DIU Take-Off Programming Contest Fall 2022 Slot A

Daffodil International University

https://toph.co/c/diu-take-off-fall-2022-slot-a

Toph

Schedule

The contest will run for **3h0m0s**.

Authors

The authors of this contest are Ahamed210, AlbinHossain, Fellow_junior, FI_Nabil, RAIKO_, samiatisha, Shefat2002, and sourav14220.

Rules

This contest is formatted as per the official rules of ICPC Regional Programming Contests.

You can use C++11 GCC 7.4, C++14 GCC 8.3, C++17 GCC 9.2, C++20 GCC 12.1, C11 GCC 12.1, and C11 GCC 9.2 in this contest.

Be fair, be honest. Plagiarism will result in disqualification. Judges' decisions will be final.

Notes

There are 8 challenges in this contest.

Please make sure this booklet contains all of the pages.

If you find any discrepencies between the printed copy and the problem statements in Toph Arena, please rely on the later.

A. LET'S START

Limits 1s, 512 MB

Hi everyone (কি অবস্থা সবার!!!). Let me tell you a small incident that happened to me. Every year a national programming contest takes place called “Ada Lovelace National Girls’ Programming Contest”. I couldn’t identify why it’s called “Ada Lovelace” (What a Shame!!!). Then I asked one of my friends. She looked at me like I committed a crime and told me nothing. So, I googled and found this -

1883: The first programming language was developed in 1883 when **Ada Lovelace** and **Charles Babbage** worked together on the Analytical Engine, which was a primitive mechanical computer. Lovelace was able to discern the importance of numbers, realizing that they could represent more than just numerical values of things. Lovelace wrote an algorithm for the Analytical Engine, the first computer program, to compute Bernoulli numbers.

Now, I can understand why she looked at me like this and I know many of you guys also don’t know who was the first to invent a programming language but Now you will remember this fact by solving this problem.

In today's world, computer programming is required to keep the systems and devices we use every day operating smoothly. Programming languages enable humans to interact with machines and make them perform necessary operations. Humans and machines process information differently, and programming languages are the key to bridging the gap between people and computers.

I have told you guys the importance of programming languages and remember **“It’s never too late until you start”.**

It is very fascinating how programming languages work and perform so fast. You are the one who is going to be one of the top programmers in the world and your journey starts from here right this moment by solving this amazing problem.

Enough telling story let’s solve this problem. All you have to do is print “**Ada Lovelace” (without quotations).**

Let me help you with the **C** program to print “**Programming is the best!!**”.

```
#include <stdio.h>

int main() {

    printf("Programming is the best!!\n");
```

```
    return 0;
}
```

Input

There is no input for this problem.

Output

Output the line mentioned in the problem statement to print.

Samples

Input	Output
	Ada Lovelace

B. Trailing Zero Offer

Limits 1s, 64 MB

Recently, Mr. Dumbo heard about a shop called "1 to 99 shop". The Prices of all the products in this shop are between 1 taka to 99 taka. The shop provides a great offer called "Trailing Zero Offer". In this offer, customers can get a significant discount on some products which have a price with trailing zeroes. Today Mr. Dumbo comes to the shop to buy some products. He picks some products to buy. Can he get the offer for these products?

A trailing zero is the zero of a number, after which no other digits follow. Consider a number n, if n is divisible by 10, the number has a trailing zero. The numbers between 1 to 99 having trailing zeros are $10, 20, 30, 40, 50, 60, 70, 80$ and 90.

Given the price of a product Mr. Dumbo wants to buy. Your task is to determine if he can get the offer for that product.

Input

The input contains only an integer $n (1 \leq n \leq 99)$ — the price of a product.

Output

If the price has any trailing zero, print `"Yes"` (without quotes). Otherwise, print `"No"` (without quotes).

Samples

Input	Output
`10`	Yes

In the test case, the price has trailing zero because $10 \% 10 = 0$, which means he can buy the product on offer.

Input	Output
`11`	No

Input	Output
In the test case, the price has no trailing zero because $11\%10 \neq 0$, which means he cannot buy the product on offer.	

Input	Output
20	Yes
In the test case, the price has trailing zero, which means he can buy the product on offer.	

Input	Output
65	No
In the test case, the price has no trailing zero, which means he can not buy the product on offer.	

C. Meal Rate

Limits 1s, 256 MB

N bachelors live in a flat and each of them eats M meals in a month. They buy groceries of T taka in a month for cooking the meals. Can you calculate the average cost per meal (meal rate) after the end of the month?

For example, 6 people live in a flat and every person eats 50 meals each in a month and they bought groceries of 15000 taka. So, the meal rate will be $\frac{15000}{6 \times 50} = 50$ taka.

Given N, M and T. Calculate the Meal Rate for them.

Input

The single line of input contains three integers $N (1 \le N \le 30)$ — the number of people living in the flat, $M (1 \le M \le 90)$ — the number of meals eaten by each person in a month and $T (1 \le T \le 30000)$ — the total amount of taka used for buying groceries in that month.

Output

Print a single integer — the meal rate after the end of the month. **It is guaranteed that output will always be an integer number.**

For more clarification checkout the note.

Samples

Input	Output
`5 50 10000`	`40`

For this sample, 5 people live in a flat and every person eats 50 meals each in a month and they bought groceries of 10000 taka. So, the meal rate will be $\frac{10000}{5 \times 50} = 40$ taka.

Input	Output
`10 60 24000`	`40`

For this sample, 10 people live in a flat and every person eats 60 meals each in a month and they bought groceries of 24000 taka. So, the meal rate will be $\frac{24000}{10 \times 60} = 40$taka.

Input	Output
6 50 15000	50
The sample is explained in the statement.	

D. Score Points by Finding Score

Limits 1s, 64 MB

Turo is playing a survival game where he has to survive some arenas by defeating all the enemies within a given time limit. Suppose Turo defeated enemies within X seconds and the time limit was Y seconds for a certain arena. Then he can clear the arena only if $X \leq Y$.

There are two arenas to win a match. Clearing each arena sequentially adds some points to the player's Total Point. Initially Turo's Total Point is 0.

The match goes as follows–

1. If Turo defeats enemies in ARENA 1 within 35 seconds (Including 35), he gets 75 points. Otherwise, he will lose and get no points.
2. After a successful ARENA 1 Match, if Turo defeats enemies in ARENA 2 within 45 seconds (Including 45), he will get 150 points more and win the match. But if he can't clear either of ARENA 1 or ARENA 2, he won't get any points for ARENA 2.

Note that, He is playing each arena independently.

Recently the game has been too slow to display the score page and Turo is desperate to know the results. You will be given **the time** that Turo managed to defeat the enemies for each arena. Your task is to determine **whether Turo is able to win or lose** the match and **print the total points**. Can you help him with the results?

Input

A single line will contain two integers t_1 and t_2 $(1 \leq t_1, t_2 \leq 100)$separated by spaces — denoting the times taken to complete ARENA 1 and ARENA 2 respectively.

Output

In the first line, print "**Win**" (without quotes), if Turo wins the match by clearing all two arenas, otherwise print "**Lose**" (without quotes), if he fails in at least one arena.

In the second line, print one integer — the total points earned by Turo throughout the match.

Check out the samples for clarification.

Samples

Input	Output
`23 37`	`Win` `225`

In this sample, Turo won the match because

- For ARENA 1 our first condition met which is 23 seconds ≤ 35 seconds. Turo's total points for now is $(0+75)=75$ points.
- For ARENA 2 our second condition met which is 37 seconds ≤ 45 seconds. Turo's total points for now is $(75+150)=225$ points.

Input	Output
`38 40`	`Lose` `0`

In this sample, Turo lost the match because

- For ARENA 1 Turo couldn't manage to defeat the enemies within 35 seconds because he took 38 seconds. Turo's total points for now is 0 points.
- Since he couldn't win ARENA 1, he won't be getting ARENA 2 winning points even if he cleared ARENA 2 and the match will be considered as a lose with total points 0.

Input	Output
`21 55`	`Lose` `75`

In this sample, Turo lost the match because

- For ARENA 1 our first condition met which is 21 seconds ≤ 35 seconds. Turo's total points for now is $(0+75)=75$ points.
- But for ARENA 2 Turo couldn't manage to defeat the enemies within 45 seconds because he took 55 seconds. Since he couldn't win ARENA 2, he won't be getting ARENA 2 winning points and the match will be considered as a lose with total points 75.

Input	Output
`36 100`	`Lose` `0`

In this sample, Turo lost the match because

- For ARENA 1 Turo couldn't manage to defeat the enemies within 35 seconds because he took 36 seconds. Turo's total points for now is 0 points.
- Since he couldn't win ARENA 1 and ARENA 2 his total points remain 0 with losing the match

Input	Output
`35 45`	`Win` `225`

In this sample, Turo wins the match because

- For ARENA 1 our first condition met which is 35 seconds $=$ 35 seconds. Turo's total points for now is $(0 + 75) = 75$ points.
- For ARENA 2 our second condition met which is 45 seconds $=$ 45 seconds. Turo's total points for now is $(75 + 150) = 225$ points.

E. Cursed Numbers

Limits 1s, 512 MB

Given two integers n and x. Consider the natural numbers from 1 to n. Some of these numbers are good numbers and others are bad. Again, if a number is divisible by x then it's an unlucky number, otherwise lucky. If a number is bad and also unlucky, then it is called a cursed number.

Now you are given for each number from 1 to n if it is good or bad. Can you count the cursed numbers between 1 and n (inclusive)?

Input

Input will contain two lines.

First line contains two integers $n, x (1 \leq n, x \leq 10^5)$ separated by space.

Second line will contain n space separated integers. Each of them is either 0 or 1. **The i-th integer is 1 if i is a bad number, and 0 if i is a good number.** You have to answer each test independently.

Output

Print a single integer — the count of cursed numbers from 1 to n.

Samples

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>6 3
0 0 1 0 1 1</pre></td><td><pre>2</pre></td></tr>
<tr><td colspan="2">For this sample, consider the six numbers from 1 to 6. On the second line the 3rd, 5th and 6th input is 1 that means 3, 5, and 6 are bad numbers. But only 3 and 6 are divisible by $x = 3$, that means 3 and 6 are unlucky numbers. So, 3 and 6 are bad as well as unlucky, hence, they are cursed numbers. So, the count of cursed numbers is 2.</td></tr>
</table>

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>5 2
1 0 1 0 1</pre></td><td><pre>0</pre></td></tr>
</table>

<u>Input</u>	<u>Output</u>
7 1 0 0 0 0 0 1 0	1

F. Division Tivision

Limits 1s, 512 MB

You are given four integers a, b, c and d. You have to determine whether $(a \times b)$ is tivisible by $(c \times d)$. Here we are not talking about usual division operation "$\div$". Rules for the tivision are:

- Let number of twos(2) as factor(s) in nominator $(a \times b)$ is x,
- Let number of twos(2) as factor(s) in denominator $(c \times d)$ is y, (It is guaranteed that value of y will be greater than or equal to 1)
- If x is divisible by y then $(a \times b)$ is tivisible by $(c \times d)$, Otherwise not.

Let's visualize an example -

Given $a = 10, b = 5, c = 2$ and $d = 7$. So $(a \times b) = 50$ and $(c \times d) = 14$

Now total number of twos in nominator

$(50 = 2^1 \times 5^2$) is 1. So, $x = 1$.

And total number of twos in denominator

$(14 = 2^1 \times 7^1)$ is 1. So, $y = 1$.

So $(x \bmod y) = (1 \bmod 1) = 0$, thus $(a \times b)$ is tivisible by $(c \times d)$.

Input

The only line of each test case contains four integers a, b, c and d.

$(0 \leq a, b \leq 1000000)$

$(1 \leq c, d \leq 1000000)$

$(c \times d)$ is even.

Output

For each test case print only line "`YES`" (without quotes) if $(a \times b)$ is tivisible by $(c \times d)$, Otherwise "`NO`" (without quotes).

Samples

Input	Output
10 5 2 7	YES

Test case 1 is explained in the statement.

Input	Output
25 25 2 1	YES

Given $a = 25, b = 25, c = 2$ and $d = 1$. So $(a \times b) = 625$ and $(c \times d) = 2$

Now total number of twos in nominator

$(625 = 2^0 \times 5^4\ )$ is 0. So, $x = 0$.

And total number of twos in denominator

$(2 = 2^1)$ is 1. So, $y = 1$.

So $(x \bmod y) = (0 \bmod 1) = 0$, thus $(a \times b)$ is tivisible by $(c \times d)$.

Input	Output
88 1 4 3	NO

Input	Output
176 1000 6767 6	YES

G. Secret Treasure

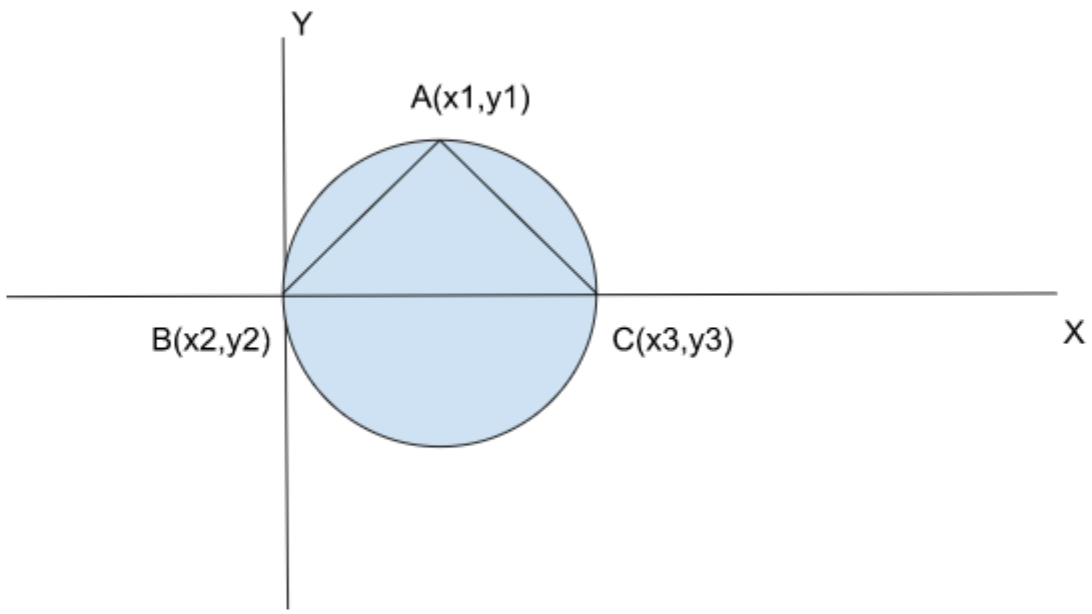
Limits 1s, 512 MB

Alice is a renowned treasure hunter. Recently he found a papyrus in a museum. There was a secret problem written on it. If he can solve this problem, he might be able to know the secret information of a new treasure. As he is not a math genius, he asked you to solve this problem.

You previously knew three important information:

1. If a, b and c are the lengths of the sides of a triangle and the area of the triangle is d then the circumradius will be $\frac{abc}{4d}$.
2. Euclidean distance between two points (x_1, y_1) and (x_2, y_2) is $(x_2 – x_1)^2 + (y_2 – y_1)^2$.
3. Area of a triangle is half of its base times height. Area = $\frac{base \times height}{2}$.

Now as shown in the following figure, given point A, and the area of $\triangle ABC$. Also stated that point B is always at the origin that is $B(x_2, y_2) = O(0, 0)$ and point C is on the positive side of the X axis. Can you find the Diameter of the circumcircle of the $\triangle ABC$?



Input

The first line will be a single integer T indicating the number of test cases. The following T lines will contain three integers x_1, y_1, and d separated by spaces. Where d denotes the area of the triangle $\triangle ABC$.

$1 \leq T \leq 100$

$1 \leq x_1, y_1 \leq 10^3$

$1 \leq d \leq 10^9$

$x_2 \leq x_1 \leq x_3$

Output

You have to output T lines in the format **"Case X: D"(without quotes)** where X is the number of test cases starting from 1 and D is the Diameter of the circumcircle of the triangle $\triangle ABC$. Your answer is considered correct if its error doesn't exceed 10^{-6}.

Samples

Input	Output
1 5 5 26	Case 1: 10.407689466006152

Input	Output
2 7 9 100 11 11 105	Case 1: 22.402900612291852 Case 2: 19.311282191333987

H. Anagrams

Limits 1s, 256 MB

A string x is called an anagram of the string y, if it is possible to rearrange letters in x so that it is identical to the string y. For example, the string "aab" is an anagram of the string "aba" and the string "aaa" is not. Another example, string “abc” has 6 anagrams: “abc”, “acb”, “bac”, “bca”, “cab” and “cba”.

A substring of string s is a continuous segment of letters from s. For example, "fod" is a substring of "daffodil" and "dof" is not. Two substrings with same content are considered different if their positions of occurrence are different. For example, the string "aba" has six substrings: "a", "b", "a", "ab", "ba", "aba". Here, the first and the third substrings are different, though their content is the same.

Now, you are given a string s consisting of lowercase Latin letters. Find another string with maximum length which is anagram to more than one substring of s. You are not required to print the actual string. Only print the maximum possible length of such a string.

Input

The first line of input data contains an integer $t (1 \leq t \leq 10000)$ —the number of test cases in the test. The descriptions of the test cases follow.

Each test case consists of one string $s (1 \leq |s| \leq 1000000)$, where $|s|$ — the length of the string s. The string consists of lowercase Latin letters.

It is guaranteed that the sum of $|s|$ on all test cases does not exceed 1000000.

Output

For each test case, print a single integer — the maximum possible length of the string which is anagram to more than one substring of s.

Samples

Input	Output
6 aba fghijklmnopqrstuvwxyz cbac abcabc abcacba acbbca	2 0 3 3 6 5

For the first test case, a possible answer string with maximum length 2 is “ab” which is anagram to two substrings “ab” and “ba” among the six substrings of given string “aba”: "a", "b", "a", "ab", "ba", "aba". It can be proven that no such string possible with size larger than 2.

For the second test case, there is no possible such string.

For the third test case a possible answer string with maximum length 3 is “abc” which is anagram to two substrings “cba” and “bac” among the ten substrings of given string “cbac”: “c”, “b”, “a”, “c”, “cb”, “ba”, “ac”, “cba”, “bac”, “cbac”. It can be proven that no such string possible with size larger than 3.