Problem A. Codeforces Rating

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

Tourist likes competitive programming and he has his own Codeforces account. He participated in lots of Codeforces Rounds, solved so many problems and became "Legendary Grand Master"(the highest rank on Codeforces). One day, he logged in to check his account. Something wrong happened: his name was in black, like he never participated in any Codeforces Rounds and his rating was incorrect. He received an e-mail from Mike Mirzayanov (the founder of the website), "we are very sorry for this mistake, everything will be fixed in few minutes"Mike said. Tourist is relaxed now. He decided to have some fun until the bug is fixed, he also would like to help the founders of the website to fix all the bugs, so he will select a random user on Codeforces, open his contests page and start to sum the rating changes. Finally, he will make sure the sum is equal to the final rating of this user.
Note: when someone creates a new Codeforces account, the initial rating is 1500.

Input

The input consists of several test cases. The first line of the input contains a single integer T, the number of the test cases.
Each test case starts with a single line containing two space-separated integers N and R: N is the number of rating changes and R is the final rating of the user.
($1 \leq N \leq 100, 0 \leq R \leq 4000$). The next line contains N space-separated integers $a_0, a_1, ..., a_{n-1}$ where $-2000 \leq a_i \leq 2000, 0 \leq i < N$ (a_i represents the change in rating in the ith contest where $0 \leq i < N$)

Output

For each test case, print a single line: "Correct"if the sum of rating changes (plus 1500) is equal to the final rating of the user and "Bug"otherwise.

Example

standard input	standard output
3 5 2000 100 100 100 100 100 2 1600 100 5 3 100 100 -1800 300	Correct Bug Correct

Problem B. Bonapity

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

A group of junior programmers are attending an advanced programming camp, where they learn very difficult algorithms and programming techniques! Near the center in which the camp is held, is a professional bakery which makes tasty pastries and pizza. It is called 'Bonabity'... or 'Ponapety'... or 'Ponabity'... Actually no one knows how to spell this name in English, even the bakery owner doesn't, and the legends say that Arabs always confuse between 'b' and 'p', and also between 'i' and 'e', so 'b' for them is just the same as 'p', and 'i' for them is just the same as 'e', they also don't care about letters' cases (uppercase and lowercase for a certain letter are similar). For example, the words 'Ponabity' and 'bonabety' are considered the same. You are given two words including only upper case and lower case English letters, and you have to determine whether the two words are similar in Arabic.

Input

The input consists of several test cases. The first line of the input contains a single integer T, the number of the test cases.
Each of the following T lines represents a test case and contains two space-separated strings (each one consists of only upper case and lower case English letters and its length will not exceed 100 characters).

Output

For each test case print a single line: 'Yes' if the words are similar in Arabic and 'No' otherwise.

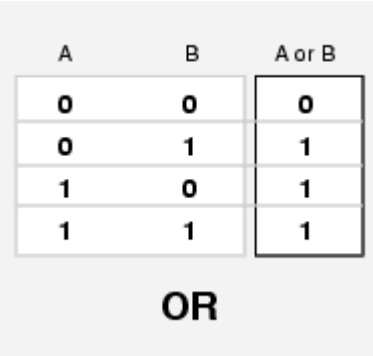
Example

standard input	standard output
4 Ponabity bonabety barbie barpee abcabc apcap abc apcd	Yes Yes No No

Problem C. A or B Equals C

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	64 megabytes

Rami went back from school and he had an easy homework about bitwise operations (and,or,..) The homework was like this : You have an equation : "A | B = C "(this bar '|' means OR) you are given the value of A and B , you need to find the value of C ?? as we said before this is an easy question for Rami to solve ,but he wonderd if the equation was like this: "A | C = B "and he is given the value of A and B , how many value of C exists ?? Rami wants your help in this hard question ... He will help you as much as he can ,so he will convert the two decimal numbers(A and B) to binary representaion (like this: if A=6 –> A=110) and this what he knows about OR operation and might be helpfull with your task :



Input

The input consists of several test cases. The first line of the input contains a single integer T, the number of the test cases. each test case consists of 3 lines : the 1st line is the length of the binary representaion of the 2 numbers.1<=n<=100 the 2nd line is the number A in base 2 .(a string consits of 0s and 1s only) the 3rd line is the number B in base 2 .(a string consits of 0s and 1s only)

Output

for each test case,you need to print the number of possible values of C that make the eqaution correct in a seperate line .(read page 2 carefully)

Example

standard input	standard output
3 2 10 11 3 110 110 4 1110 1011	2 4 IMPOSSIBLE

Note

as the answer may be very very large , you should print the answer mod 1000000007. there might be no answer for the equation also , in this case you should print "IMPOSSIBLE"(without qoutes).

Problem D. X and paintings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

X is well known artist, no one knows the secrete behind the beautiful paintings of X except his friend Y, well the reason that Y knows X's secrete is that he is that secret. Y is a programmer and he helps X with drawing paintings using computer program written by him. Unfortunately Y program is not working any more, now it's your turn to help X by writing a program that helps him in painting, your program should accept a sequence of instructions, each will draw an opaque (filled) rectangle in the form:
r_1, c_1, r_2, c_2, *color*. where (r_1, c_1) is the upper left corner of the rectangle and (r_2, c_2) is the lower right corner, and *color* is a character that denotes the color of this rectangle. all rectangles will be printed on a R by C plane, by default this plane is filled with dots (i.e. '.'). R and C between 1 and 100. for each instruction ($1 \leq r_i \leq R$) and ($1 \leq c_i \leq C$) and *color* is a ASCII character [a-z]. The number of instructions won't exceed 100 instructions.

Input

The first line of input contains an integer T denotes number of test cases.
The first line of each test case contains three integers R, C ,I where R and C denotes number of rows and columns of the painting, and I denotes number of instructions. Each of the next I lines contains four integers r_1,c_1,r_2,c_2 and the *color* character.

Output

Print the final plane after evaluating the instructions in order.

Example

standard input	standard output
1 5 5 3 1 1 2 2 a 1 2 5 5 c 2 2 3 3 d	acccc addcc .ddcc .cccc .cccc

Problem E. Teams

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

ACM-SCPC-2017 is approaching every university is trying to do its best in order to be the Champion, there are n universities, the i^{th} of them has exactly a_i contestants. No one knows what is the official number of contestants in each team yet, this year organizers are planing to minimize number of teams participating in the contest but they still want every contestant to participate, so they should select a number k so that each team will consist of exactly k contestant from the same university. Every contestant will belong to one team exactly, and number of these teams is as minimum as possible. Your job is to help the organizers in their task by writing a program that calculates two integers k, the number of contestants in each team, and m, the minimum number of teams will participate.
$1 \leq n \leq 1000$, $1 \leq a_i \leq 10^6$

Input

First line of input contains an integer T denotes number of test cases. Each test case contains two lines: the first line contains one integer n denotes number of universities, while the second line contains n space-separated integers the i^{th} of them denotes number of contestants from the i^{th} university.

Output

For each test case, print one line containing two integers: the first one is k, the size of each team, and the second one is m, the minimum number of teams from all universities according to the previous conditions.

Example

standard input	standard output
2 3 5 15 10 4 4 6 8 12	5 6 2 15

Problem F. Good Words

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

Zaid has two words, a of length between 4 and 1000 and b of length 4 exactly. The word a is 'good' if it has a substring which is equal to b. However, a is 'almost good' if by inserting a single letter inside of it, it would become 'good'. For example, if a = 'start' and b = 'tear': b is not found inside of a, so it is not 'good', but if we inserted the letter 'e' inside of a, it will become 'good' ('steart'), so a is 'almost good' in this case.
Your task is to determine whether the word a is 'good' or 'almost good' or neither.

Input

The input consists of several test cases. The first line of the input contains a single integer T, the number of the test cases.
Each of the following T lines represents a test case and contains two space separated strings a and b, each of them consists of lower case English letters.
It is guaranteed that the length of a is between 4 and 1000, and the length of b is exactly 4.

Output

For each test case, you should output one line: if a is 'good' print 'good', if a is 'almost good' print 'almost good', otherwise print 'none'.

Example

standard input	standard output
4 smart mark start tear abracadabra crab testyourcode your	almost good almost good none good

Note

A substring of string s is another string t that occurs in s.
Let's say we have a string $s =$ "$abcdefg$"
Possible valid substrings: "a","b","d","g","cde","$abcdefg$".
Possible invalid substrings: "k","ac","$bcef$","dh".

Problem G. The Tower of Evil

Input file:	`evil.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

The forces of evil are about to disappear since our hero is now on top on the tower of evil, and all what is left is the most evil, most dangerous monster!
The tower has h floors (numbered from 1 to h, bottom to top), each floor has w rooms (numbered from 1 to w, left to right) composing a row. The monster stands in one of the ground floor (floor number 1) rooms (the room number d where $1 \leq d \leq w$). Our hero stands in the top-left room of the tower.
The only way for our hero to kill the evil monster is to throw down one of his power stones diagonally to the right. The stone will keep moving diagonally through the tower's rooms until it hits the right border of the tower, then it will change its direction to move down diagonally to the left until it hits the left border of the tower, then it will change its direction again and so on.
This stone stops when it reaches the ground floor, if it stopped in the room of the monster, the monster is dead, otherwise the monster is still alive and the forces of evil will rise again!
Help our hero to determine whether the stone will kill the monster or not!

Input

The input consists of several test cases. The first line of the input contains a single integer T, the number of the test cases.
Each of the following lines contains a test case and consists of a three space-separated integers h, w and d denoting the height of the tower, the width of the tower and the number of room containing the monster. $(1 \leq d \leq 10^9)$, $(2 \leq h, w \leq 10^9)$.

Output

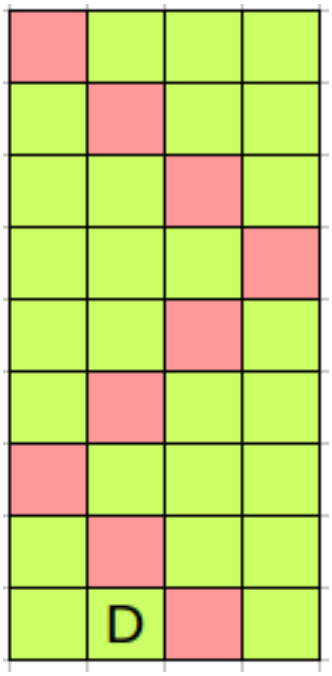
For each test case print a single line: 'Yes' if the monster will be killed and 'No' otherwise.

Example

evil.in	standard output
5 9 4 2 9 4 3 5 4 3 10 2 1 10 2 2	No Yes Yes No Yes

Note

In the first test case, the path of the power stone is colored in red. However the dragon is in the room denoted with D, so the monster is still alive and the answer is 'No'.


D

Problem H. The Endless River

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

You've possibly heard about 'The Endless River'. However, if not, we are introducing it to you. The Endless River is a river in Cambridge on which David and Roger used to sail. This river has the shape of a circular ring, so if you kept sailing forward you wouldn't find an end, and that's why it was called 'endless'. The river is exactly n meters long. Each minute, David moves d meters forward while Roger moves r meters forward; they start sailing from the same position at the same time, moving in the same direction. At the beginning of each minute, each of them leaves a marker at his current place (they don't put any markers at the beginning of the race). Determine the number of minutes before two markers (of different people) are placed at the same position.

Input

The input consists of several test cases. The first line of the input contains a single integer T, the number of the test cases.
Each of the following lines represents a test case and contains three space-separated integers n, d and r ($1 \leq n, d, r \leq 100000$) denoting the length 'in meters' of the river, David's speed and Roger's speed (as explained in the statement above) respectively.

Output

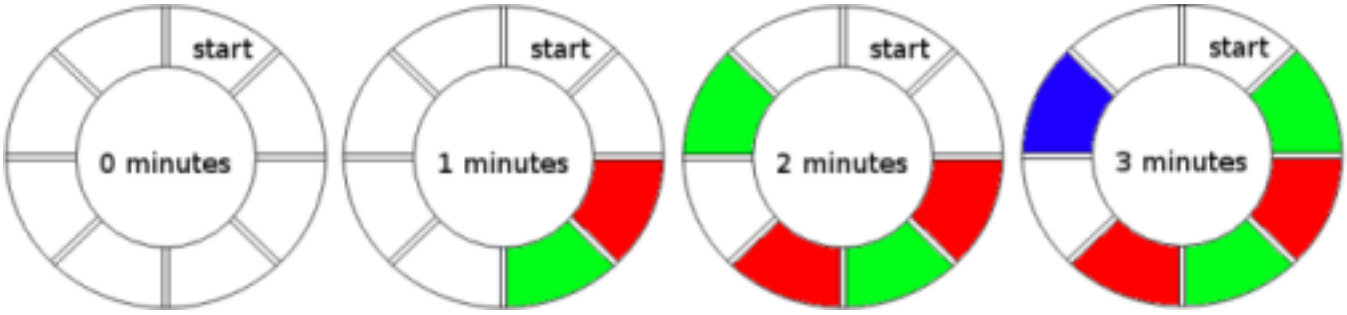
For each test case print a single line containing one integer: the number of minutes before two markers (of different people) are placed at the same position.

Example

standard input	standard output
4 8 2 3 8 3 2 5 1 4 100 1 1	3 3 3 1

Note

In the first case, we have $n = 8$, $d = 2$ and $r = 3$. They start at the first space which is denoted with 'start'. Red color denotes David's markers, green for Roger's markers, blue denotes that two markers (of different people) are placed.
At the beginning (0 minutes passed), no markers are placed.
After 1 minute passes, David is at the third space and Roger is at the fourth.
After 2 minutes pass, David is at the fifth space and Roger is at the seventh.
After 3 minutes pass, David is at the seventh space and Roger is at the minute, so when David places his marker, two markers are placed at the seventh space and the answer is 3 minutes.


start
0 minutes
start
1 minutes
start
2 minutes
start
3 minutes

Problem I. March Rain

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

It is raining again! Youssef really forgot that there is a chance of rain in March, so he didn't fix the roof of his house. Youssef's roof is 1-D, and it contains n holes that make the water flow into the house, the position of hole i is denoted as x_i where $(0 \leq i < n)$. Youssef has to put strips at the bottoms of those holes in order to prevent the water from flowing. Let's say there is a hole in position 4 and another hole in position 6, and Youssef decided to use a strip of length 3 to cover those two holes, then he places the strip from position 4 to 6 (it covers positions 4,5,6) and it covers the two holes. He can buy exactly k strips, and he must pay a price equal to the longest strip he buys. What is the minimum length l he can choose as the longest strip in order to keep his house safe?

Input

The input consists of several test cases. The first line of the input contains a single integer T, the number of the test cases.
Each test case consists of two lines: the first line contains two space-separated integers, n and k $(1 \leq k < n \leq 100000)$, denoting the number of the holes in the roof, and the number of the strips he can buy respectively. The second line of the test case contains n integers $(x_0, x_1, ..., x_{n-1})$: $(0 \leq x_i \leq 10^9)$, denoting the positions of holes (these numbers are given in an increasing order).

Output

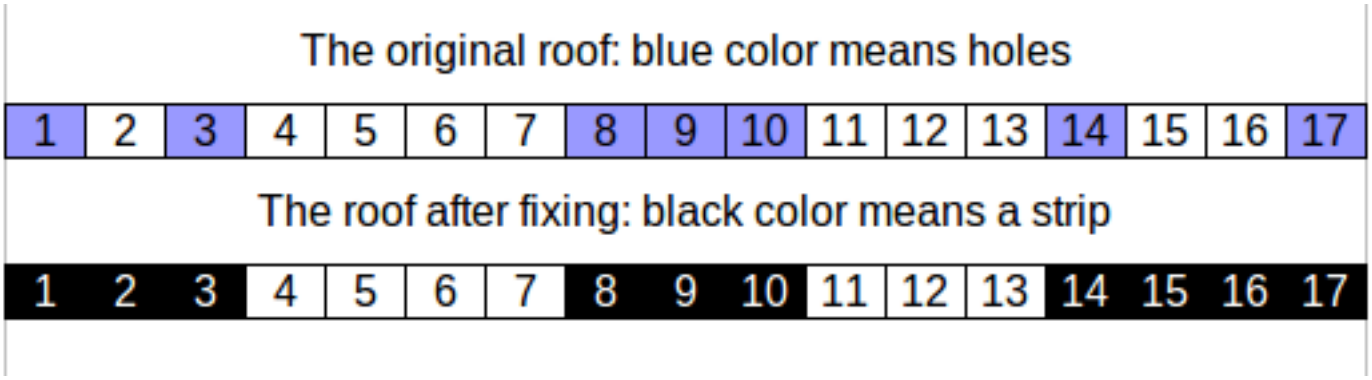
For each test case print a single line containing a single integer denoting the minimum length l he can choose in order to buy k strips (the longest of them is of length l) and cover all the holes in his house using them.

Example

standard input	standard output
3 5 2 1 2 3 4 5 7 3 1 3 8 9 10 14 17 5 3 1 2 3 4 20	3 4 2

Note

In the second test case the roof looks like this before and after putting the strips.



Problem J. X and Beasts

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

X is fighting beasts in the forest, in order to have a better chance to survive he's gonna buy upgrades for his weapon. Weapon upgrade shops are available along the forest, there are n shops, where the i^{th} of them provides an upgrade with energy a_i.
Unfortunately X can make use only of the maximum power of two that divides the upgrade energy, lets call it the *powerincrement*. For example, an upgrade with energy of 6 has a power increment of 1 because 6 is divisible by 2^1 but not divisible by 2^2 or more, while for upgrade with energy of 5 power increment is 0, also after buying a weapon with energy v, X can only buy upgrades with *energies* greater than v, in other words, the upgrades energies that X is gonna buy should be in strictly increasing order. X is wondering what is the maximum power he can achieve at the end of his journey. note that only the energy of upgrades should be in strictly increasing order **not** the power increment of the upgrade.
$1 < n < 100$, $1 \leq a_i \leq 10^6$

Input

The first line of input contains one integer, T, which denotes number of test cases.
Each test case contains two lines, the first one contains one integer n which denotes number of shops, and the next line contains n integers the i^{th} of them denotes the energy of the upgrade provided by the i^{th} shop.

Output

Print one integer denotes maximum power X can achieve at the end of his journey

Example

standard input	standard output
2 3 1 10 16 2 8 12	5 5