The International Collegiate Programming Contest
Sponsored by ICPC Foundation

EGYPTIAN COLLEGIATE PROGRAMMING CONTEST

The 2021 ECPC Contest

(Contest Problems)

AAST, Egypt
August 2021

Problem A. Intern Judges

Input file:	`intern.in`
Output file:	`standard output`
Balloon Color:	`Red`

Endure Capital is an early stage investment fund headed by entrepreneurs. They have been one of the main sponsors of the ECPC, and they are always curious to know the number of problems at the ECPC. They know judges won't reveal the number of problems, so they sent a message to Eng. Fouad to ask for the number of problems at ECPC 2021.

As you may know, judges contribute to the problem-set with n problems. However, this year, a new role at the ECPC was added. "Intern Judges" are new judges which are still under training. As the ECPC community is encouraging everyone to participate in the problem setting, intern judges were asked to add exactly 2 problems to the problem set.

Your task is easy, help Mr. Fouad calculate the total number of problems so he can report to Endure, given that judges contribute with n problems and intern judges contribute with exactly 2 problems.

Input

The first line contains a single integer T ($1 \leq T \leq 100$), the number of test cases.

Each test case contains a single line containing a single integer n ($1 \leq n \leq 50$), the number of problems provided by judges.

Output

For each test case, print a single integer, the total number of problems provided by judges and intern judges.

Example

intern.in	standard output
3 10 34 13	12 36 15

Problem B. Fuel Problem

Input file: `fuel.in`
Output file: `standard output`
Balloon Color: `Black`

Coach Academy is one of ACPC partners. They train hundreds of contestants in African and Arab countries every year. They teach problem solving and competitive programming to teens, university students and graduates. When they heard of the ECPC Teens, they decided to take their young problem solvers to Alexandria to participate in the contest.

Coach Academy rented 2 buses to take the contestants to the AAST campus. Unfortunately, as they were in a hurry, the bus drivers forgot to fuel the 2 buses, and they managed to get only K liters of fuel for the 2 buses. They have decided to split it between them.

The capacity of Bus 1 tank is A liters, while the capacity of Bus 2 tank is B liters. They want to distribute the K liters of fuel between them in the fairest way possible. We say the unfairness of distribution is the absolute difference between the two amounts Bus 1 and Bus 2 have taken.

Note, that it's possible to fill the tank of the bus with any real number of liters, not just an integer amount. Obviously, each bus can't take any amount more than the capacity of his tank and the total amount of K liters must all be distributed.

Find a way to distribute the fuel with the minimum possible unfairness.

Input

The first line of the input contains a single integer number T $(1 \leq T \leq 10^3)$ – the number of test cases.

Each test-case is described with a single line containing three space-separated integers K, A and B $(1 \leq K, A, B \leq 200)$, $(K \leq A + B)$.

Output

For each test-case, output a single line containing the minimum possible unfairness.

Example

fuel.in	standard output
2 1 1 1 31 10 46	0 11

Problem C. Volunteers

Input file:	volunteer.in
Output file:	standard output
Balloon Color:	Gold

Volunteering at the ECPC is one of the most interesting jobs you could ever do. You get to choose between a number of fascinating teams to join. Teams include media, systems, scientific committee, operations and more. Volunteers also get to socialize, know ALOT of new people and make new friends.

Sometimes, volunteers play games while performing their tasks to kill the time. One of these games is as follows. Two volunteers are randomly chosen. One of them chooses a string S and the other chooses a string R. The rest of the volunteers try to generate all the strings that are lexicographically strictly greater than R using any character in S at most once (They can leave some characters).

Not all volunteers are good at problem solving, so they have decided to ask you to generate the strings for them.

A string $S1$ is called lexicographically strictly greater than a string $S2$ if there exists an index i where $S1_j = S2_j$ for every $j < i$ and $S1_i > S2_i$.

Input

The first line contains an integer T denoting the number of test cases.

For each test-case:

- The first line of each test-case contains string S.
- The following line contains string R.

The length of each string doesn't exceed 10^5 characters.

It's guaranteed that all strings consist of lowercase English letters only and the sum of lengths of S and R over all test cases doesn't exceed 10^6.

Output

For each test case, print the maximum number of strings formed from S that are lexicographically greater than R in a single line. Then, print the strings, each in a single line in any order.

Example

volunteer.in	standard output
3 abcd b ddccbb dca xyz xy	3 ba c d 2 dcb dcb 2 y z

Problem D. Math Lovers

Input file: `math.in`
Output file: `standard output`
Balloon Color: `Violet`

There are ALOT of math loverss at the ECPC. Bolbol and Balabeelo are two of them. When it's time for a break between the practice session and the contest, they use it to play games involving math. Today, they are playing the following game:

The game starts with a sequence A of N positive integers written on a piece of paper, players alternate turns and Bolbol starts first. In one turn, the player must do one of the following moves:

1- select an integer bigger than 1 and replace it by 1.

2- select an integer bigger than 1 and replace it by all its divisors except itself.

The player who can't move in his turn loses. Your task is to determine who wins if both players play optimally.

Input

The first line of the input contains a single integer number T - the number of test cases.

For each test case:

- The first line of each test case contains a single integer number N $(1 \le N \le 10^6)$ - the number of elements of the sequence A.
- The following line contains N space-separated integer numbers $A_1, A_2, ..., A_N$ $(1 \le A_i \le 10^9)$, where A_i is the i_{th} element of the sequence A.

It's guaranteed that the sum of N over all test cases doesn't exceed 10^6.

Output

For each test case, print a single line containing "Bolbol" if Bolbol will win. Otherwise, print "Balabeelo" -without quotes-.

Example

math.in	standard output
4 3 2 6 12 5 3 1 4 1 5 7 2 7 1 8 2 8 1 11 1 6 1 8 3 3 9 8 8 7 5	Bolbol Bolbol Bolbol Bolbol

Problem E. Good Strings

Input file: `good.in`
Output file: `standard output`
Balloon Color: `Velvet`

As usual, when the judges can't think of a name for some string/array, they call it "good". Here is another one of those problems:

You are given two strings s, t and q queries. Each query consists of four integers l and r and i and j. To answer this query, you should count the number of occurrences of string z in s where z is the concatenation of $t_{l,r}$ and $t_{i,j}$, where $t_{l,r}$ is the sub-string of string t starting with l and ending with r. Simple right?

For example, if s is "abababa" and t is "ababa", the answer to the query $l = 2$, $r = 3$, $i = 2$, $j = 2$ is the number of occurrences of the string "bab" in s which is equal to 2.

Input

The first line contains a single integer T, the number of test cases.

Each test case starts with a line containing the string s $(1 \le |s| \le 10^5)$.

The following line contains the string t $(1 \le |t| \le 10^5)$.

The following line contains a single integer q $(0 \le q \le 10^6)$ – the number of queries.

Each of the following q lines contains four space-separated integers l, r $(1 \le l \le r \le |t|)$ and i, j $(1 \le i \le j \le |t|)$

It is guaranteed that the sum of $|s| + |t|$ over all test cases does not exceed 10^6.

It is also guaranteed that the sum of q over all test cases does not exceed 10^6.

Output

For each test case, print q lines each containing a single integer, the answer to the respective query.

Example

good.in	standard output
1 abababa ababa 3 2 3 2 2 1 1 3 3 1 5 4 5	2 0 1

Problem F. First to Solve

Input file: nodes.in
Output file: standard output
Balloon Color: Light Green

Today, you are helping the balloon runners at the ECPC. The contest floor is modelled as a weighted tree, with n nodes that have $n-1$ edges. Nodes represent teams, and special nodes are those teams which got 1st accepted solutions, so balloon runners have to deliver balloons to without passing by any other special nodes, so they don't get jealous. 1st to solve teams are all distinct.

You want to visit every special node (team with a 1st accepted) at least once, to deliver the balloons. To do so, you can repeat the following procedure any number of times:

Assuming you are currently at node u,

- Select any node v (not necessarily special) such that there isn't any special node along the path from u to v exclusively (i.e, not considering u and v), and go to node v.
- Add the maximum weight among all visited edges to the total cost.
- Change the weight of all visited edges to 0.

Initially, you are at node 1. Your task is to find the minimum total cost to visit every special node at least once.

Input

The first line contains an integer T denoting the number of test cases.

For each test case:

- The first line contains an integer n ($1 \le n \le 10^5$) – the number of nodes in the tree.
- The next $n-1$ lines contain three integers u, v and w ($1 \le u, v \le n$), ($1 \le w \le 10^9$) which means there is an edge between node u and v that weights w.
- The next line contains an integer k ($1 \le k \le n$) – the number of special nodes.
- The next line contains k space-separated integers $a_1, a_2, ..., a_k$ ($1 \le a_i \le n$) – the special nodes.

It's guaranteed that the sum of n over all test cases doesn't exceed 2×10^5.

Output

For each test case, print the minimum total cost to visit all special nodes.

Example

nodes.in	standard output
1 6 1 2 9 2 3 5 3 4 3 3 5 2 3 6 5 2 5 6	11

Problem G. No Cheating

Input file:	`cheat.in`
Output file:	`standard output`
Balloon Color:	`Green`

They look like bees, don't they? All those yellow t-shirts around you are in fact the contest floor watchmen. One of their duties is to make sure no team is cheating from another. Some contestants, unfortunately, will try to cheat. There are n teams in the classroom, where team i will be looking at team P_i's PC the whole time. Note, that there might be some i such that $P_i = i$, which means that the team i is honest and won't cheat. We define the *reachability* of team i as the number of **other** teams they might get solutions from.

For example, if $P = [2, 3, 3, 5, 4]$.

- The 1^{st} team has a reachability of 2 (they can reach teams 2 and 3)
- The 2^{nd} team has a reachability of 1 (they can reach team 3)
- The 3^{rd} team has a reachability of 0 (they are only looking at their PC)
- The 4^{th} and 5^{th} teams, each has a reachability of 1 (they are looking to each others PC)

Yellow volunteers can not prevent everyone from cheating, They will keep an eye on one team only and stop them from cheating. They want your help to choose such team in a way that minimizes the sum of *reachability* of all teams.

It's enough to print the final sum of reachability.

Input

The first line of the input contains a single integer number T, the number of test cases.

Each test case contains two lines:

- The first line contains one integer n $(1 \le n \le 10^5)$, the number of teams.
- The second lines contains n integers $P_1, P_2, ..., P_n$ $(1 \le P_i \le n)$.

It's guaranteed that the sum of n over all test cases doesn't exceeded 5×10^5.

Output

For each test case, output a single line containing one integer, the final sum of reachability of all teams.

Example

cheat.in	standard output
1 5 2 3 3 5 4	3

Problem H. Shifts

Input file:	`shift.in`
Output file:	`standard output`
Balloon Color:	`Yellow`

ECPC has witnessed a great growth in the past couple of years, especially this year. The number of teams has jumped from 600+ teams in 2020 to 1400 teams in 2021. Teams are becoming stronger and have more knowledge and so problems are becoming more interesting, like this one.

Given an array a of length n, an empty array b and an integer number m which initially equals 0. You can do the following operation infinite number of times:

- Append all elements in a to the end of b in the same order.
- Assign $m := m + 1$.
- Do m cyclic left shifts on a.

You will then be asked q questions. In each question, we want to know the value of the k-th index in b. Can you help us find the answer for all the questions?

Input

The first line of the input contains a single integer number t - the number of test cases.

For each test case:

- The first line of each test case contains a single integer number n ($1 \leq n \leq 10^5$) - the number of elements in a.
- The second line of each test case contains n space-separated integer numbers a_i ($1 \leq a_i \leq 10^9$), where a_i is the i-th integer in a.
- The third line of each test case contains a single integer q ($1 \leq q \leq 10^5$) - the number of questions that you will be asked.
  - Each of the next q lines contains a single integer number k ($1 \leq k \leq 10^9$). The index in the array b that you must find.

It is guaranteed that the sum of n over all of the test cases will not exceed 10^5.

It is guaranteed that the sum of q over all of the test cases will not exceed 10^5.

Output

For each testcase, for each question, print a single line containing a single integer number. The k-th number in b.

Example

shift.in	standard output
2 3 1 2 3 9 1 2 3 4 5 6 7 8 9 5 3 3 3 3 3 3 7 59 3	1 2 3 2 3 1 1 2 3 3 3 3

Problem I. Letter Cubes

Input file:	`cube.in`
Output file:	`standard output`
Balloon Color:	`Rose`

Judges at the ECPC got bored of reviewing wrong answer submissions, so they decided to buy letter cubes and create a game with them.

Each cube has an English letter written on it. They counted all the numbers of letter cubes of each letter of the English alphabet to obtain the sequence: $F_0, F_1, ..., F_{25}$ of 26 integers, where F_0 is the number of letter cubes which have the letter 'a' written on it, F_1 is the number of letter cubes which have the letter 'b' written on it and so on.

Judge Ali Magdy loves palindromes so much; that's why he wondered; what's the maximum number of palindrome strings with distinct lengths that can be formed using these letter cubes. Each cube can be used at most once.

A palindrome is a string that reads the same backward as forward. For example, strings "z", "aaa", "aba", "abccba" are palindromes, but strings "accepted", "reality" and "ab" are not.

Input

The first line contains an integer T $(1 \le T \le 10^4)$ denoting the number of test cases.

For each test case, you are given a line containing 26 space-separated integers $F_0, F_1, ..., F_{25}$. $(0 \le F_i \le 10^{16})$.

Output

For each test case, print a single line containing an integer denoting the maximum number of palindrome strings with distinct lengths that can be formed using the given letter cubes.

Example

cube.in	standard output
3 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	26 4 3

Problem J. Yosri's Fortune

Input file:	`fortune.in`
Output file:	`standard output`
Balloon Color:	`Orange`

The systems team at the ECPC is one of the most hard-working teams ever. All the PCs, networking, and onsite support are managed by them. They share the room with the judges, so the 2 teams (judges and systems) are always in contact.

One day, judge Yosri was explaining the modulo to Sara, one of the system team members at the ECPC. She got so confident with it that she solved any modulo problem given to her by Yosri. He decided to challenge her with a harder problem and promised to give her half his fortune if she solved it.

Yosri gave Sara an array a of length n, an integer m and an integer k. He wants her to find the number of subsequences b in a modulo $10^9 + 7$ such that:

- The length of b is divisible by k.
- $b_1 \equiv b_2 \equiv \ldots b_k \bmod m$.
- $b_{k+1} \equiv b_{k+2} \equiv \ldots b_{2k} \bmod m$.
- and so on.

More formally, the following condition must hold: $b_{ik+1} \equiv b_{ik+2} \equiv \ldots b_{ik+k} \bmod m$.

Can you help Sara solve this problem and have half of Yosri's fortune?

Input

The first line of input contains a single integer number t - the number of testcases.

For each test case:

- The first line of each testcase contains three space-separated integer numbers n, m and k $(1 \le n, m \le 10^5, 1 \le k \le 100)$.
- The second line of each testcase contains n space-separated integer numbers $(0 \le a_i \le 10^5)$, where a_i is the i-th number in a.

It is guaranteed that the sum of n over all of the testcases does not exceed 5×10^5.

Output

For each testcase, print a single line containing a single integer, the number of subsequences in a that hold the two conditions modulo $10^9 + 7$.

Example

fortune.in	standard output
3 2 2 1 1 1 4 5 2 0 8 10 13 6 2 2 1 1 2 1 2 2	3 2 11

Problem K. Weird Forumla

Input file: `weird.in`
Output file: `standard output`
Balloon Color: `Cyan`

One of the best things about qualifying to the ACPC, is socializing with other contestants, coaches and judges. One day at the last ACPC (Luxor 2020), Safrout the chief judge gave Coach Fegla an interesting problem.

Safrout gave Fegla two sequences $A_1, A_2, ..., A_N$ and $B_1, B_2, ..., B_N$, both of which have length N. He wanted Fegla to select a non-empty subsequence of indices $1 \le i_1 < i_2 < ... < i_k \le N$ to maximize

$$\frac{\text{INV}}{B_{i_1} \text{ OR } B_{i_2} \text{ OR} \cdots \text{OR } B_{i_k}}$$

Where INV is the number of pairs (p, q) such that $1 \le p < q \le k$ and $A_{i_p} > A_{i_q}$, and OR is the bitwise OR operator.

Although Fegla can solve the problem without any help, he is asking you to tell him the maximum value the expression can get, for a chance to join his senior training.

Input

The first line of the input contains a single integer number $T (1 \le T \le 5)$ - the number of test cases.

For each test case:

- The first line of each test case contains a single integer number N $(1 \le N \le 2000)$ - the length of sequence A and B.
- The following line contains N space-separated integer numbers $A_1, A_2, ..., A_N$ $(1 \le A_i \le 10^9)$, where A_i is the i_{th} element of the sequence A.
- The following line contains N space-separated integer numbers $B_1, B_2, ..., B_N$ $(1 \le B_i \le 10^6)$, where B_i is the i_{th} element of the sequence B.

Output

For each test-case, output a single line containing two integers - the answer in an irreducible fraction format, where the first number is the numerator and the second one is the denominator.

Example

weird.in	standard output
2 5 5 1 4 2 3 3 1 4 2 1 7 7 3 4 1 5 2 6 3 4 2 7 1 1 5	1 1 5 3

Problem L. High Walls

Input file: walls.in
Output file: standard output
Balloon Color: Purple

Do you see the gray t-shirts all around you? Yes, they are the operations team, the joker of the ECPC. They set up the contest halls, judging rooms, balloon zone and everything in between.

They have made a wall around the contest hall to eliminate the noise, but made a hole in it to allow balloon runners to enter and leave the contest hall.

The wall is simply a circle with a radius r meters and a center $C(0,0)$. The hole made the wall becomes an arc with a gap starting at angle a and ending at angle b. Note, that point $(r,0)$ is considered to be at angle 0, and we are measuring angles counterclockwise.

The balloon runners are currently at position $T(t_x, t_y)$, and they will take the shortest path to reach the center of the contest hall. They move one meter per second.

Help the runners by calculating how much time they have before they reach the center of the contest hall.

It's guaranteed that the point $T(t_x, t_y)$ doesn't lie on the Circle.

Input

The first line of the input contains a single integer number T ($1 \le T \le 10^5$), the number of test cases.

Each test case is described with a single line containing five integers r, a, b, t_x, t_y, ($1 \le r \le 10^7$), ($-10^7 \le t_x, t_y \le 10^7$), ($0 \le a < b < 360$).

Output

Output a single real number for each test case - how many seconds before the balloon runners reach the center of the contest hall.

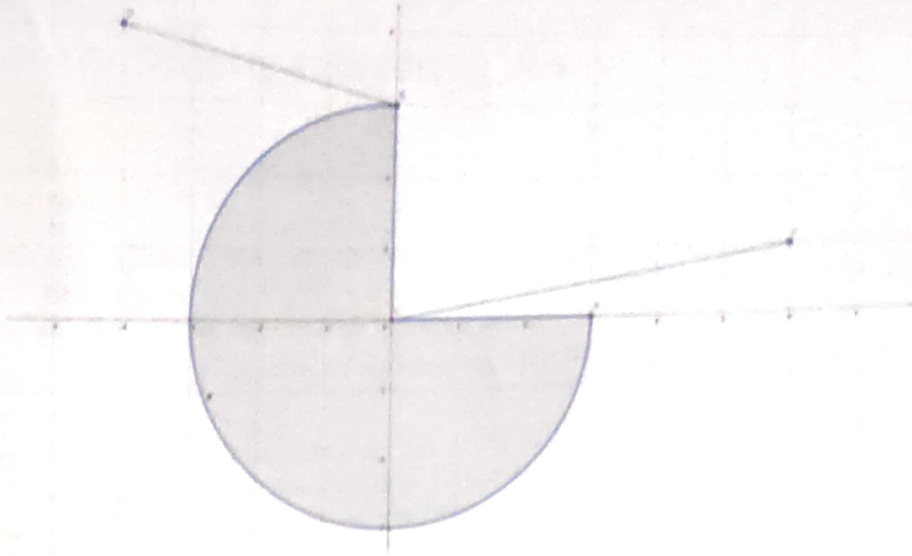
Your answer is considered correct if its absolute or relative error doesn't exceed 10^{-4}. Namely, if your answer is a, and the jury's answer is b, then your answer is accepted, if $\frac{|a-b|}{\max(1,b)} \le 10^{-4}$

Example

walls.in	standard output
2 3 0 90 -4 4 3 0 90 6 1	7.123106 6.082763

Note

The first and second samples only differ by the balloon runners' positions



In the first sample, balloon runners will go from point D to B then C.

In the second sample, balloon runners can go directly from F to C.

Problem M. Strictly Increasing Array

Input file:	`inc.in`
Output file:	`standard output`
Balloon Color:	`Pink`

In their continuous efforts to enhance the performance of Egyptian problem solvers across the country, the scientific committee at the ECPC holds sessions and trainings for old and new Egyptian ICPC communities, accommodating from newbie to senior levels.

You have been a part of the committee's training, and you found a very interesting problem to solve in the weekly training sheet.

The problem provides you with an array $A_1, A_2, ..., A_N$, and then you start modifying a sub-segment of this array by selecting an index l which points to the first element of this sub-segment and an index r that points to its end. The first element of the sub-segment is incremented by 1 and the second element is incremented by 2... and so on until the last element of the sub-segment is incremented by $r - l + 1$.

More formally, after selecting the desired sub-segment, it will be changed as follows:

- $A_l = A_l + 1$,
- $A_{l+1} = A_{l+1} + 2$,
- $A_{l+2} = A_{l+2} + 3$,
- $... = ... + ..,$
- $A_r = A_r + r - l + 1$

Your task is to find the minimum number of sub-segment modifications on the array for it to become strictly increasing. Will you be able to do so, or do you need to attend another training?

One of the students tried to modify his program to be able to calculate that minimum answer, but he did not succeed in doing so, so he asked you to help him completing this task.

Input

The first line contains an integer T denoting the number of test cases.

For each test case:

- The first line of each test case contains a single integer N $(1 \leq N \leq 10^5)$ denoting the length of the array.
- The next line contains N integers A_i $(1 \leq A_i \leq 10^9)$.

It's guaranteed that the sum of N over all test-cases doesn't exceed 10^5.

Output

For each test-case, print on a separate line the minimum number of subsegments that need to be selected to make the array A strictly increasing.

Example

inc.in	standard output
3 3 1 2 2 5 1 22 333 4444 55555 5 1 2 1 2 1	1 0 2

Problem N. Equal Array

Input file:	`mulrange.in`
Output file:	`standard output`
Balloon Color:	`Blue`

You are given an array a of n integer numbers. You can perform the following operation any number of times:

- Choose three integer numbers l, r and k $(1 \le l \le r \le n, 1 \le k)$.
- Multiply each element of a within the range $[l, r]$ by k. More formally, for each index $(l \le i \le r)$, apply the operation $a_i = a_i \times k$. performing such operation costs k.

The total cost is the sum of all operations' costs, calculate the minimum possible total cost to make all elements of a equal.

Input

The first line of the input contains a single integer number T. The number of test cases.

The first line of test case contains a single integer number n $(1 \le n \le 10^6)$. The number of elements in the array a.

The following line contains N space-separated integer numbers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^6)$, where a_i is the i_{th} element in the array a.

It's guaranteed that the sum of n over all test cases doesn't exceed 10^6.

Output

For each test case print a single integer number, the minimum cost needed to make all the elements in the array a equal.

Example

mulrange.in	standard output
3 3 2 4 8 6 2 3 1 6 8 4 9 3 1 4 1 5 9 2 6 5	4 17 35

Note

In the first test case, $a = [2, 4, 8]$.

In the first operation, choose $(l = 1, r = 2, k = 2)$ with cost 2. The array becomes $a = [4, 8, 8]$

In the second operation, choose $(l = 1, r = 1, k = 2)$ with cost 2. The array becomes $a = [8, 8, 8]$.

The minimum possible cost is $2 + 2 = 4$.

Problem O. Strong Password

Input file:	`password.in`
Output file:	`standard output`
Balloon Color:	`White`

Brimore is so far the biggest social commerce app in Egypt and Africa. It allows people to set up their virtual shops and share thousands of products. They recently hired you as a security engineer to check if their user accounts are secure enough.

It's believed that an account is secure if it's password is strong. If the password is weak, the the account is not secure enough.

A password is strong only if it has at least 10 characters. Otherwise, it is considered weak.

Brimore will give you some password strings. Your job is to determine whether each password is strong or weak.

Input

The first line of the input contains a single integer number T, that doesn't exceed 30, the number of test cases.

Each test-case contains a single line containing a string S. The length of S doesn't exceed 50, and it's guaranteed to consist of only small English letters, capital English letters or digits.

Output

For each test-case, output one of the following words "weak" or "strong" -without the quotes- depending on the password.

Example

password.in	standard output
3 12345678 IamWeak7 IamStro0ong012	weak weak strong