TLE 11.0 Level 1 - Contest 1

A. Sum

2 s., 512 MB

Priyam and his friends are participating in a math challenge. Each of them is given a simple addition problem where they have to quickly find the sum of two single-digit numbers.

Your task is to help them by evaluating each given expression and providing the result.

Input
The first line contains one integer t ($1 \le t \le 100$) — the number of test cases.

Each test case consists of one line containing an expression of the form $a+b$ ($0 \le a, b \le 9$, both a and b are integers). The integers are **not** separated from the $+$ sign.

Output
For each test case, print one integer — the result of the expression.

input
4 4+2 0+0 3+7 8+9
output
6 0 10 17

B. GCD game

1 s., 256 MB

Dhruvil is a young mathematician who loves to explore number properties. He's particularly fascinated by the concept of greatest common divisors (GCD). One day, he decides to play a game with his friend Tushar using the first n positive integers.

The game works as follows:

- Dhruvil writes down all integers from 1 to n (inclusive) on a blackboard.
- Tushar then chooses any two **different** numbers from the board.
- Dhruvil calculates the GCD of these two numbers.

Dhruvil wants to know the highest possible score (GCD) that Tushar can achieve in this game. Can you help Dhruvil determine this maximum score?

Remember, the greatest common divisor (GCD) of two positive integers is the largest positive integer that divides both numbers without a remainder.

Input
The first line contains a single integer t ($1 \le t \le 100$) — the number of games Dhruvil and Tushar want to play.

For each game: The only line contains a single integer n ($2 \le n \le 10^6$) — the upper limit of the range of numbers written on the blackboard.

Output
For each game, output the maximum GCD that Tushar can achieve by choosing any two different numbers from 1 to n.

input
2 3 5
output
1 2

In the first game where $n = 3$, the possible pairs and their GCDs are: $\mathrm{GCD}(1, 2) = 1$, $\mathrm{GCD}(1, 3) = 1$, $\mathrm{GCD}(2, 3) = 1$. So the maximum GCD Tushar can achieve is 1.

In the second game where $n = 4$, Tushar can achieve a maximum GCD of 2 by choosing the pair $(2, 4)$, as $\mathrm{GCD}(2, 4) = 2$.

C. Card Game

1 s., 256 MB

Priyam and his friends are playing a card game in the fictional city of Luminoria. There are $2n$ cards on the table, each with a positive integer written on it. Priyam, being the last player, gets to play a special round.

Priyam starts with 0 points. He will increase his score by performing the following move **exactly** n times:

- Choose two cards from the table.
- Add the lower of the two numbers on the chosen cards to his score.
- Remove both cards from the table.

After Priyam has made n moves, all cards will be removed from the table. Help Priyam determine the maximum score he can achieve by playing optimally.

Input
The input consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 5000$) — the number of test cases.

For each test case: The first line contains a single integer n ($1 \le n \le 50$) — half the number of cards on the table. The second line contains $2n$ integers $a_1, a_2, \ldots, a_{2n}$ ($1 \le a_i \le 10^7$) — the numbers written on the cards.

Output
For each test case, output the maximum score Priyam can achieve.

input
3 1 2 3 2 1 1 2 1 3 1 1 1 1 1 1
output
2 2 3

In the first test case, Priyam can only make one move. He selects cards with values 2 and 3, adding the lower value (2) to his score.

In the second test case, Priyam can achieve a maximum score of 2 by:

- First, selecting two cards with 1, adding 1 to his score.
- Then, selecting the remaining cards with 1 and 2, adding 1 to his score.

In the third test case, Priyam will make three moves, adding 1 to his score each time.

D. Number Taste

1 s., 256 MB

Priyam has a unique taste in numbers. He dislikes any number that is divisible by 3 or ends with the digit 3 in their decimal representation. If a number meets both conditions, Priyam dislikes it even more.

Priyam starts writing out the positive (greater than 0) integers which he likes: $1, 2, 4, 5, 7, 8, 10, 11, 14, 16, \ldots$. Your task is to find the k-th element of this sequence (the elements are numbered from 1).

Input
The first line contains one integer t ($1 \le t \le 100$) — the number of test cases. Then t test cases follow.

Each test case consists of one line containing one integer k ($1 \le k \le 1000$).

Output
For each test case, output in a separate line one integer x — the k-th element of the sequence that was written out by Priyam.

input
10 1 2 3 4 5 6 7 8 9 1000
output
1 2 4 5 7 8 10 11 14 1666

E. Energy Sequences

2 s., 256 MB

In a futuristic world where data is stored as vast energy sequences, two explorers, Priyansh and Priyam, have discovered two massive energy patterns, labeled e_a and e_b. Your task is to compare these two energy sequences and determine which one holds more energy, or if they are equal.

These sequences are extremely large, and because of their size, you cannot analyze them one character at a time. Instead, you must read the entire sequence at once using highly efficient input and output methods. For instance:

- In C++, use `scanf/printf` instead of `cin/cout`.

- In Java, use `BufferedReader/PrintWriter` instead of `Scanner/System.out`.

- In Python 2, use `raw_input()` rather than `input()`.

Input
The first line contains the energy sequence e_a (a non-negative integer with possible leading zeroes).

The second line contains the energy sequence e_b (a non-negative integer with possible leading zeroes).

Both e_a and e_b have at most 10^6 digits.

Output
Print the symbol "<" if $e_a < e_b$, the symbol ">" if $e_a > e_b$, or "=" if the energy sequences are equal.

input
9 10
output
<

input
11 10
output
>

input
00012345 12345
output
=

input
0123 9
output
>

input
0123 111
output
>

F. Yet yet yet another treasure hunt

1 s., 256 MB

In an underground treasure hunt competition, gems are distributed as follows:

- Whenever a team finds a treasure chest, that team receives a gem.
- The first team to open a specific treasure chest receives an extra gem.

The treasure hunt has 26 hidden treasure chests, labelled A, B, C, ..., Z. You are given the sequence in which the treasure chests were found by various teams, denoted by a string s. In this string, the i-th character indicates that treasure chest s_i has been found by a team. No team can find the same chest more than once.

Your task is to determine the total number of gems distributed in the competition. Note that some treasure chests might not be found by any team.

Input
The first line of the input contains an integer t ($1 \le t \le 100$) — the number of test cases.

The first line of each test case contains an integer n ($1 \le n \le 50$) — the length of the string.

The second line of each test case contains a string s of length n consisting of uppercase English letters, representing the sequence in which treasure chests were found.

Output
For each test case, output a single integer — the total number of gems the teams received.

input
6 3 ABA 1 A 3 ORZ 5 BAAAA 4 BKPT 10 CODEFORCES

output
5 2 6 7 8 17

In the first test case, 5 gems are distributed:

- Treasure chest A is found. That team receives 2 gems: one for finding the chest and an extra one for being the first team to find treasure chest A.
- Treasure chest B is found. That team receives 2 gems: one for finding the chest and an extra one for being the first team to find treasure chest B.
- Treasure chest A is found again. That team only receives 1 gem, as they are not the first team to find treasure chest A.

The total number of gems distributed is $2 + 2 + 1 = 5$.
In the second test case, only one chest is found, and the team that finds it receives 2 gems: one for finding the chest and an extra one for being the first team to find treasure chest A.

G. Coolant flowing

1 s., 256 MB

In a distant research outpost on a barren planet, a scientist named Nova is attempting to direct a stream of liquid coolant to her malfunctioning reactor core. The coolant delivery system, however, was designed for cooling multiple reactors and consists of a pipe with n nozzles, each representing a leak point where coolant escapes. Nova can only utilize the coolant flowing out of the **first nozzle** to stabilize her reactor.

Nova has the ability to block some of the nozzles and then inject A liters of coolant into the system. The coolant flows out of the unblocked nozzles in proportion to their sizes $q_1, q_2, \ldots, q_n$. Specifically, if the total size of the unblocked nozzles is Q, and the i-th nozzle is not blocked, then $\frac{q_i \cdot A}{Q}$ liters of coolant will flow out of that nozzle.

Nova's goal is to ensure that at least B liters of coolant flow from the **first nozzle**. What is the minimum number of nozzles Nova must block to achieve this?

Input
The first line contains three integers n, A, and B ($1 \le n \le 100\,000$, $1 \le B \le A \le 10^4$) — the number of nozzles, the volume of coolant injected into the system, and the minimum volume required to flow out of the first nozzle.

The second line contains n integers $q_1, q_2, \ldots, q_n$ ($1 \le q_i \le 10^4$) — the sizes of the nozzles.

Output
Print a single integer — the number of nozzles Nova needs to block.

input
4 10 3 2 2 2 2

output
1

input
4 80 20 3 2 1 4

output
0

input
5 10 10 1000 1 1 1 1

output
4

In the first example, Nova must block at least one nozzle. Afterward, $\frac{10 \cdot 2}{6} \approx 3.333$ liters of coolant will flow out of the first nozzle, which meets Nova's requirement.

In the second example, even without blocking any nozzle, $\frac{80 \cdot 3}{10} = 24$ liters of coolant will flow out of the first nozzle, which is sufficient.

In the third example, Nova must block all nozzles except the first to direct all coolant to the malfunctioning reactor.

H. Energy level 3

1 s., 256 MB

You are tasked with managing a group of highly specialized nanobots, each programmed to collect a specific amount of a rare energy source, represented by an array $e = [e_1, e_2, \ldots, e_n]$, where each e_i is the energy collected by the i-th nanobot.

In one operation, you can merge the energy collected by any two nanobots into a single nanobot and remove the selected two nanobots from the array and the new nanobot with an energy level equal to the sum of the two chosen nanobots' energies can be placed anywhere in the array. For example, if your group has energy levels $[2, 1, 4]$, after an operation, the group could have the configurations $[3, 4]$, $[1, 6]$, or $[2, 5]$.

Your objective is to maximize the number of nanobots whose energy level is a multiple of 3 after performing any number of these merging operations (including zero).

Input
The first line contains one integer t ($1 \le t \le 1000$) — the number of missions.

The first line of each mission contains one integer n ($1 \le n \le 100$).

The second line of each mission contains n integers $e_1, e_2, \ldots, e_n$ ($1 \le e_i \le 10^9$).

Output
For each mission, print one integer in a single line — the maximum possible number of nanobots with energy levels divisible by 3 after performing the described operations any number of times.

input
2 5 3 1 2 3 1 7 1 1 1 1 1 2 2

output
3 3

In the first mission of the example, you can apply the following sequence of operations to obtain 3 nanobots with energy levels divisible by 3:
$[3, 1, 2, 3, 1] \to [3, 3, 3, 1]$

In the second mission, you can achieve 3 nanobots with energy levels divisible by 3 by using this sequence of operations: $[1,1,1,1,1,2,2] \to [1,1,1,1,2,3] \to [1,1,1,3,3] \to [2,1,3,3] \to [3,3,3]$.

I. Musical Night

2 s., 256 MB

Priyansh is organizing a unique musical performance where musicians are arranged in a line. He has n musicians, each with their own rhythm value. Priyansh wants to divide the musicians into two groups: those who will play drums and those who will play flutes. To create an interesting performance, he needs to ensure that no two adjacent musicians play the same instrument.

To decide who plays what, Priyansh chooses a special rhythm number d. Musicians whose rhythm value is divisible by d will play drums, while the others will play flutes. Your task is to help Priyansh find any value of d that creates a valid arrangement, or determine if it's impossible to do so.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of performances to plan.

For each performance:

The first line contains one integer n ($2 \le n \le 100$) — the number of musicians. The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^{18}$), where a_i is the rhythm value of the i-th musician.

Output

For each performance, print a single integer. If there's no value of d that creates a valid arrangement, print 0. Otherwise, print any suitable value of d ($1 \le d \le 10^{18}$).

input
5 5 1 2 3 4 5 3 10 5 15 3 100 10 200 10 9 8 2 6 6 2 8 6 5 4 2 1 3
output
2 0 100 0 3

In the first test, if Priyansh chooses $d = 2$, then the 2nd and 3rd musicians will play drums, while the 1st, 3rd, and 5th musicians will play flutes.

J. Energy Grid

2 s., 256 MB

Priyansh is exploring a high-tech energy grid on an alien planet. The grid is of size $n \times n$, and there are m energy orbs that he can place on the grid. Initially, the grid is empty, but Priyansh will place the orbs one by one, and the goal is to analyze how much of the grid remains unaffected by the energy.

Each orb emits an energy field that affects all cells in the same row and column. If there is an orb placed in a cell, that cell is also considered affected by the energy.

You are given the positions where Priyansh will place each orb. After placing each orb, your task is to determine how many cells in the grid remain unaffected by the energy.

Input

The first line contains two integers n and m ($1 \le n \le 100,000$, $1 \le m \le \min(100,000, n^2)$) — the size of the grid and the number of orbs.

Each of the next m lines contains two integers x_i and y_i ($1 \le x_i, y_i \le n$), which represent the row and column where Priyansh places the i-th energy orb. Priyansh places the orbs in the order they appear in the input. It is guaranteed that no cell will contain more than one orb.

Output

Print m integers. The i-th integer should represent the number of cells that are unaffected after placing the first i energy orbs.

input
3 3 1 1 3 1 2 2
output
4 2 0

input
5 2 1 5 5 1
output
16 9

input
100000 1 300 400
output
9999800001

After each orb is placed, certain rows and columns become affected by the energy emitted. The remaining cells that are unaffected after each placement need to be counted, just like in the image where the grey areas represent unaffected cells.