Intra Metropolitan University Programming Contest - (IMUPC) 2025

Problem Set

CoderOJ

A. The Card Clash Problem

Time: 1 s
Memory: 125 MB

In a strange card game, each team is represented by a lineup of players. Each player either holds up a card or keeps it down. The team's formation is represented by a binary string of equal length for all teams, where 1 indicates a card is up and 0 indicates a card is down.

During the game, teams face off in matches. When two teams play a match, they simultaneously compare and update their formations based on the following rule:

For each position in the formation (from right to left):

- If both players at that position have their cards up , both set their cards down.
- Otherwise, the players swap their card states at that position — the player with the card down sets it up, and the one with the card up sets it down.

This transformation is applied for every match, in the given order. After all matches are complete, each team's score is the decimal value of its final binary formation.

Your task is to determine the top 3 teams based on their scores. If multiple teams have the same score, the team that appeared earlier in the input ranks higher.

Input

- The first line contains a single integer T — the number of test cases.
- Each test case begins with two integers N and X — the number of teams and the number of bits in each team's formation.
- The next N lines each contain a string name and a binary string formation of exactly X bits — representing the team name and its initial formation.
- Then follows an integer M — the number of matches.
- Each of the next M lines contains two strings team1 and team2 — the names of two teams that play a match.

Constraint

$1 \leq T \leq 1000$

$1 \leq N \leq 10^5$

$1 \leq X \leq 60$

$1 \leq M \leq 10^5$

$Sum\ of\ N\ over\ all\ testcases\ \ \leq 10^5$

$Sum\ of\ M\ over\ all\ testcases\ \ \leq 10^5$

$1 \leq Length\ of\ team\ names \leq 30$

Output

For each test case, output 3 lines. Each line should contain the rank, team name, and final score in the following format:

$1.\ TeamName\ Score$
$2.\ TeamName\ Score$
$3.\ TeamName\ Score$

Examples

Input	Output
1 3 4 RedTeam 1011 BlueTeam 1100 GreenTeam 0101 2 RedTeam BlueTeam BlueTeam GreenTeam	1. RedTeam 4 2. BlueTeam 4 3. GreenTeam 2

Input	Output
1 4 5 Alpha 10101 Beta 11000 Gamma 01110 Delta 00111 3 Alpha Beta Gamma Delta Alpha Gamma	1. Gamma 8 2. Delta 8 3. Beta 5

Notes

Explanation for Sample 1

Initial Formations:
RedTeam: 1011
BlueTeam: 1100
GreenTeam: 0101
Match 1 (RedTeam vs BlueTeam):
Before
ReadTeam 1011
BlueTeam 1100
After
RedTeam: 0100 (4)
BlueTeam: 0011 (3)

Match 2 (BlueTeam vs GreenTeam):
Before
BlueTeam: 0011
GreanTeam: 0101
After
BlueTeam: 0100 (4)
GreenTeam: 0010 (2)
Final Scores:
RedTeam: 4
BlueTeam: 4
GreenTeam: 2

B. For the Love of One

Time: 1 s
Memory: 125 MB

Batman is obsessed with the number 1, especially when the **bitwise XOR** of all bits in a number's binary representation equals 1.
Alfred Pennywise wants to gift Batman a number, n, for his birthday. However, if the XOR of the bits in n's binary form isn't equal to 1, Alfred must perform a **bitwise OR** between n and the **smallest positive integer x** such that the result **does** satisfy Batman's obsession.
However, Alfred is very busy, so he asked for your help in finding the value of x.

Input

The input contains integer n, the number chosen by Alfred

Constraint

$1 \leq n \leq 2^{30}$

Output

Find the smallest number x so that when you OR n with x, the XOR of all bits becomes 1.
If n already has the XOR of bits equal to 1, print 0.

Examples

Input	Output
5	2

Input	Output
15	16

Input	Output
1073741824	0

Notes

In the first test case,

- Binary of 5 = 101 → 1 ^ 0 ^ 1 = 0 (not valid)
- Try x = 1:
  5 | 1 = 101 | 001 = 101 → 1 ^ 0 ^ 1 = 0 (not valid)
- Try x = 2:
  5 | 2 = 101 | 010 = 111 → 1 ^ 1 ^ 1 = 1 (valid)

So, the smallest x that makes (n | x) valid is 2.

In the second test case,
Binary of 15 = 1111 → 1 ^ 1 ^ 1 ^ 1 = 0 (not valid)
if x = 16,
15 | 16 = 1111 | 10000 = 11111 → 1 ^ 1 ^ 1 ^ 1 ^ 1 = 1 (valid)
There is no number less than 16 that can satisfy Batman's obsession. And so the answer is 16.

C. A Sweet Rivalry

Time: 1 s
Memory: 125 MB

One day, Koushik was on his way to university when he unexpectedly met his two cousins, Prohor and Bibek. Koushik was carrying several types of chocolates, each with different sweetness values. Before continuing his journey, he gave these chocolates to his cousins.

Happy with their new chocolates, Prohor and Bibek returned home and arranged them carefully in a row. In this arrangement, **sweetness[i]** represents how sweet the **i-th** chocolate is when counted from the **left end**. They decided to play a game with the following rules:

Players take turns, with **Prohor starting first**.

On a turn, a player must:

1. Choose an integer **x > 1** (removing at least 2 chocolates from the left).
2. **Remove** the leftmost **x** chocolates and **add** their total sweetness to their score.
3. Place a new chocolate (with sweetness equal to the **sum** removed) at the **leftmost** position.

The game ends when only **one chocolate** remains.

Prohor aims to **maximize** the difference (his score — Bibek's score), while Bibek aims to **minimize** this difference, return the final score **difference** if both play **optimally**.

Input

The first line contains a single integer **n** ($2 \le n \le 10^5$) — the number of chocolates. The next line contains n space-separated **sweetness[i]** represents the sweetness of the **i-th** chocolates from the **left**.

Constraint

$2 \leq n \leq 10^5$
$-10^4 \leq Sweetness[i] \leq 10^4$

Output

Print a single number — the score **difference** between Prohor and Bibek if they both play **optimally.**

Examples

Input	Output
7 7 -6 5 10 5 -2 -6	13

Input	Output
6 -10 -8 1 2 -12 3	-3

Notes

Test Case 1:
Prohor removes all chocolates, adds 7 + (-6) + 5 + 10 + 5 + (-2) + (-6) = 13 to his score, and places a chocolate of value 13 on the left. chocolates = [13].
The difference between their scores is 13 - 0 = 13.

Test Case 2:
Prohor removes the first 5 chocolates, adds (-10) + (-8) + 1 + 2 + (-12) = -27 to his score, and places a chocolate of value -27 on the left. chocolates = [-27,3].

- Bibek removes the first 2 chocolates, adds (-27) + 3 = -24 to his score, and places a chocolate of value -24 on the left. chocolates = [-24].

The difference between their scores is (-27) - (-24) = -3.

D. Circle of Harmony

Time: 1 s
Memory: 125 MB

You are given a set of n distinct elements labeled from 1 to n, arranged in a circle in clockwise order.
Define a function $f(k)$ as the number of ways to partition this set into exactly k non-empty groups such that :

- Each element belongs to **exactly one group**.
- For each group, if you draw straight line segments (chords) connecting every pair of elements within that group inside the circle,
- Then, no two chords from **different** groups intersect each other inside the circle.

Two ways of partitioning are considered different if there exists at least one element that belongs to different groups in the two partitions.
Your task is to compute the value of : $\sum_{k=1}^{n} f(k)$

—that is, the total sum of $f(k)$ for every **k** from **1 to n**.
Since the answer can be very large, output the result modulo **998244353**

Input

The first line contains an integer **T** — the number of test cases.
Each of the next **T** lines contains a single integer **n** — the size of the set.

Constraint

$1 \leq T \leq 10^5$
$1 \leq n \leq 10^6$

Output

For each test case, output a single line containing one integer — the answer to the given test case

Examples

Input	Output
3 1 3 5	1 5 42

Input	Output
3 9 24 15	4862 172443248 9694845

E. Luckiest Unlucky?

Time: 1 s
Memory: 125 MB

Meet Miss Lavender, a peculiar woman with an eye for numbers. Every day on her way to the office, she takes her usual route. Along the way, she collects exactly **2 numbers** that seem to appear mysteriously on the signs she passes.

Once she collects the numbers, she finds all the **common divisors** between them and **sums them up**. That sum holds the key to her day's fate:

- If the sum is a **prime number**, she considers herself **Lucky**.
- If the sum is **not a prime**, but **1 less or 1 more than a prime number**, she calls herself **Luckiest Unlucky**.
- Otherwise, she feels **Unlucky**.

Can you determine what kind of day Miss Lavender will have based on the numbers she collects?

Input

A single line containing two space-separated integers **a** and **b** — the numbers Miss Lavender collects.

Constraint

1 ≤ a, b ≤ 1000

Output

Print one of the following depending on the result:

- "Lucky" — if the sum of common divisors is a prime number.
- "Luckiest Unlucky" — if the sum is not a prime, but sum - 1 or sum + 1 is a prime.
- "Unlucky" — otherwise.

Print without the **quotation** marks.

Examples

Input	Output
6 16	Lucky

Input	Output
5 15	Luckiest Unlucky

Input	Output
8 24	Unlucky

Notes

In testcase 2,
Divisors of 5 = {1, 5}, Divisors of 15 = {1, 3, 5, 15}, Common divisors = {1, 5} → Sum = 6.
6 is not prime, but 5 and 7 (±1) are primes → **Luckiest Unlucky**.

F. Choose the best node

Time: 1 s
Memory: 125 MB

You are given a tree with N nodes numbered from 0 to N-1. The tree is undirected. Each edge has a positive integer weight.
Each node u has:

- An integer weight W[u] (1 ≤ W[u] ≤ 1000).
- A color C[u], which is either 0 or 1.

Let's say, $F(u) = \sum_{\substack{0 \le v < N \\ v \neq u,\, C[v]=C[u]}} \mathrm{dist}(u, v) \times W[v]$

Here, dist(u, v) denotes the sum of edge weights along the shortest path from node u to node v in the tree.

Your task is to find a node u such that F(u) is **maximized.**

Input

In the first line, there will be an integer N (1 ≤ N ≤ 100000).

In the second line, there will be given N integers; i-th of the N integers denote the weight of node-i. And in the next line, there will be given N more integers; i-th of the N integers denote the color of node-i.
The next N−1 lines each contain three integers u (0 ≤ u < N), v (0 ≤ v < N) and w (1 ≤ w ≤ 1000), meaning there is an edge between node u and node v, with edge weight w.

Note: It is guaranteed that the input represents a valid tree.

Output

Print the node **u** that maximizes **F(u)**. If there are multiple such nodes with the maximum value, find the one with the smallest node number.

Examples

Input	Output
4 3 5 9 7 1 0 0 0 0 1 29 0 2 21 2 3 13	1

Notes

For the given test case,
F(0) = 0 (since there is no node with color 1)
F(1) = 891
F(2) = 341
F(3) = 432
F(1) is the maximum of all.

G. Yet Another Spinning Spinners

Time: 1 s
Memory: 250 MB

Mr. Toto discovered a magical piece of glass. At first glance, it appears completely ordinary. However, when three pieces of this glass overlap, the overlapping region changes color and begins to glow. Fascinated by this phenomenon, Mr. Toto began thinking of ways to utilize it. After much thought, he decided to build a structure called the Spinning Spinners.

The complex spinner consists of six individual spinners, all arranged symmetrically on a circular stand. Each spinner is a circular disc made of the magical glass, with a wheel at its center. All spinners are of equal size, and the distance between the centers of any two adjacent spinners is the same. The circular stand also has the same radius(**R**) as the spinners.

Your task is to calculate two things:
1) The total area where the magical glass is glowing.
2) The area where the magical glass appears normal.

Note: Use π = **3.1416** for all calculations.

Input

The first line of input contains an integer **T**, Number of test cases. Each test case contains a single integer **R**(Radious of the spinners)**.**

Constraint

T(1<=T<=10^5)
R(1<=R<=10^8)

Output

For each test case, print the case number and two space separated numbers (the area that the glass is glowing followed by the area where the glass looks ordinary with **4 digits** after the decimal point). Your answer will be considered correct if its absolute or relative error does not exceed 10^{-4}

Examples

Input	Output
3 2 5 100000000	Case 1: 4.3482 41.5692 Case 2: 27.1762 259.8076 Case 3: 10870475772933678.0000 103923048454132640.0000

H. Abida’s Eternal CR7 Obsession

Time: 1 s
Memory: 500 MB

Abida is a die-hard fan of Cristiano Ronaldo — the legendary footballer famously known as **CR7**. As part of her endless obsession, she has started seeing the digit **'7'** as a lucky sign everywhere. One day, she challenged herself to count how many times the digit **'7'** appears in all non-negative integers from **0** to **N**, where **N** is *massive* — possibly with up to 10^{10000}.
Clearly, Abida can’t do this manually. She’s asking for your help to compute the answer efficiently.
To keep the number manageable, return the final count **modulo** $10^9 + 7$.

Input

The first line contains an integer T (1 ≤ T ≤ 10^5) — the number of test cases.
Each of the next T lines contains a single string N (1 ≤ |N| ≤ 10^4) — a non-negative integer without leading zeros.

It is guaranteed that the sum of |N| over all test cases doesn't exceed 10^6. Here |N| is number of digit in number N.

Output

For each test case, output a single integer — the total number of times the digit **'7'** appears in all numbers from 0 to N, **modulo** $10^9 + 7$.

Examples

Input	Output
3 13 77 100	1 16 20

I. Sneaky Snaky Portals

Time: 1 s
Memory: 125 MB

We've all played the classic snake game, where the snake moves inside a 2d grid. When the snake eats an apple, it grows by 1 and our score increases by 1. The game ends when the snake collides with its own body or the grid's border. Initially, the snake has a length of 1.
To make things more interesting, the game developers introduced portals in the grid and changed the scoring system. These portals allow the snake to teleport from one cell to another.
You will be given a sequence of the snake's moves. Your task is to determine the fate of the snake!
Rules:

- The snake moves according to the given sequence. The snake's body follows its head.
- If the snake's head moves onto a cell containing an entry portal, it instantly teleports to the corresponding exit portal. Its head's new position becomes the exit portal cell and continues moving from there.
- If the snake lands on a cell containing food, it eats the food, grows by 1 unit and earns 5 points. The effect occurs before the next move.
- The snake dies (Game Over!) if its head moves outside the grid or into any part of its own body.
- The starting cell of the snake, all portal cells and food cells will be distinct.

Input

The first line contains two integers n and m – the number of rows and columns in the grid.
The second line contains two integers rs and cs – the starting position of the snake's head.
The third line contains a single integer p – the number of portal pairs.
Each of the next p lines contains four integers r1, c1, r2, c2 – representing a one-way portal from cell (r1, c1) to cell (r2, c2).
The next line contains a single integer f – the number of food items.
Each of the next f lines contains two integers – the position of Grow Food. No food will be placed on a portal cell.
The last line contains a string s – a sequence of moves. Each character is one of 'U', 'D', 'L', 'R', corresponding to Up, Down, Left and Right.

Constraint

3 <= n, m <= 100
1<= rs, rf <=n
1<= cs, cf <= m
0 <= p <= 20
3 <= r1, c1, r2, c2 <= 1000
0 <= f <= 50
1 <= |s| <= 1000

Output

If the snake dies at move x, output: "Game Over! The snake died at Move x!", otherwise, after completing all moves, output: "The player's final score is y!" – where y is the final score.

Examples

Input	Output
5 5 1 1 1 1 2 4 4 1	The player's final score is 5!

Input	Output
4 3 RDLU	

Input	Output
6 6 3 3 2 4 3 1 3 1 5 5 5 3 4 2 1 4 1 6 DRRRRUR	Game Over! The snake died at Move 5!

J. Echoes from the Lab

Time: 1 s
Memory: 125 MB

The **IMUPC 2025** has just begun at **Metropolitan University**, and the entire computer lab is buzzing with excitement.
As part of the opening ceremony, the contest system was designed to echo a **motivational chant** throughout the lab — a symbolic message to fire up every programmer's spirit. But due to a minor glitch, the echo system isn't working properly. It only echoes once, when it should have echoed **multiple times**, based on how many participants are in each lab section.
You, the system wizard, have been called in to fix it. You're given the number of echoes required for each section. Your task is to **print the motivational chant exactly N times**, loud and clear.

Input

-The first line contains a single integer **T,** the number of lab sections (test cases).
-The next **T** lines each contain an integer **N**, the number of times the chant should echo in that section.

Constraint

$1 \le T \le 100$
$1 \le N \le 100$

Output

For each test case, print the string **Keep calm and code on!** exactly **N** times, each on a new line.
After each test case, print an empty line to separate sections.

Examples

Input	Output
2 3 1	Keep calm and code on! Keep calm and code on! Keep calm and code on! Keep calm and code on!