International Collegiate Programming Contest
MCPC 2020 Qualifications Round
Virtual
December 6th 2020

The International Collegiate Programming Contest
Sponsored by ICPC Foundation

MCPC 2020 Qualifications Round

(Contest Problems)

Virtual

December 2020

Problem A. Split and Swap

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

You are given two strings s and t, each string consists of n lowercase English letters.

You can perform the following operation on string s:

- Split the string into two parts and swap their order. For example, if s is ‘‘`flybutter`’’, we can split it into ‘‘`fly`’’ and ‘‘`butter`’’, then after swapping their order the string s will become ‘‘`butterfly`’’,

Your task is to find the minimum number of required `Split and Swap` operations to make string s equal to string t. **It is always possible to achieve that for the given input**.

Input

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

The first line of each test case contains a single integer n ($1 \leq n \leq 10^5$), the length of each of the strings.

Each of the following two line contains a string of n lowercase English letters, representing string s and t, respectively.

It is guaranteed that it is possible to make string s equal to string t using zero or more `Split and Swap` operations.

The total sum of n over all test cases doesn't exceed 2×10^6.

Output

For each test case, print a single line with the minimum number of required `Split and Swap` operations to make string s equal to string t.

Example

swap.in	standard output
2 9 flybutter butterfly 4 mcpc mcpc	1 0

Problem B. Game Over

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

Game Over is a multi-player point based game of N players. At the beginning everyone has zero points, after each round exactly one person gets exactly one point.

Suppose we have 3 players with a final result of $2 - 0 - 1$

There are three scenarios to reach this result:

1. Player 1 got 2 consecutive points then player 3 got a point.
2. Player 1 got a point, player 3 got a point and then player 1 got another point.
3. Player 3 got a point and then player 1 got 2 consecutive points.

However if the game ended $0 - 0 - 0$ or $5 - 0 - 0$ then there is only one possible scenario. You're given the final result of a game of Game Over, output the number of possible scenarios.

Input

First line of input will contain $T \leq 50$ number of test cases.

Each test case will start with a positive integer $2 \leq N \leq 6$, followed by N integers, the final score of each player, each player will not be able to score more than 4 points.

Output

For each test case, output the number of possible scenarios.

Example

game.in	standard output
3 3 2 0 1 3 2 1 0 3 0 4 0	3 3 1

Problem C. Cutting strings

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

Given 2 strings S_1 and S_2 consisting of lower case English letters. You need to cut S_1 into one or more consecutive non overlapping substrings, so that each of them occur in S_2. In how many ways can you do so ?

Input

The first line of the input contains a single integer $1 \leq T \leq 50$ the number of test cases. Each test case consists of two lines, the first line is S_1 and the second line is S_2. $1 \leq |S_1|, |S_2| \leq 10^5$.

Output

For each test case output a single line containing the answer of the problem modulo $10^9 + 7$.

Example

cut.in	standard output
3 aaa a aaa aa aaa aaa	1 3 4

Note

In the first case, the only possible way of cutting is [‘a’, ‘a’, ‘a’]. In the second case, the possible ways are [‘a’, ‘a’, ‘a’], [‘aa’, ‘a’] and [‘a’, ‘aa’]. In the Third case, the possible ways are [‘a’, ‘a’, ‘a’], [‘aa’, ‘a’], [‘a’, ‘aa’] and [‘aaa’].

Problem D. A High Dimensional Game

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

Saad and Diaa are playing a game on a hyper-rectangle. A hyper-rectangle is a generalization of a rectangle to higher dimensions. In 2-space, a hyper-rectangle is just a rectangle; in 3-space, a hyper-rectangle is a box. In general, one can specify a hyper-rectangle by giving the side-lengths along each of the dimensions. 2×2, 5×1, and 1673×194 are examples of 2-dimensional hyper-rectangles; $3 \times 1 \times 9 \times 5$ is an example of a 4-dimensional hyper-rectangle.

In this way, a hyper-rectangle in N dimensions can be specified by the numbers $D_1, \ldots, D_N$. A point on a hyper-rectangle is an ordered tuple of coordinates. The coordinates along the first dimension are $0, 1, 2, \ldots, D_1$; the coordinates along the second dimension are $0, 1, 2, \ldots, D_2$ and so on.

Saad and Diaa put a stick-man on the point with coordinates $(0, 0, \ldots, 0)$ and decided to move it to the point with coordinates $(D_1, D_2, \ldots, D_n)$. In one move, the next to play chooses any dimension among the N dimensions and moves the stick-man along that dimension by any positive number of steps (the stick-man can be moved forwards only) and without causing the stick-man to fall off the hyper-rectangle. The last player to make a move wins this game.

Saad plays first, then Diaa plays and they keep alternating moves until one of them wins. Given that both players play optimally, your task is to find out who wins this game. Print "SAAD"if Saad wins and print "DIAA"if Diaa wins.

Input

The first line of input contains the number of test cases T ($T \leq 10$). The first line of each test case contains the number of dimensions N. The second line contains N ($1 \leq N \leq 100$) space-separated integers D_i ($i = 1 \ldots N$) where D_i is the side length along the i-th dimension ($1 \leq D_i \leq 2 * 10^6$).

Output

For each test case print, on a separate line, SAAD if Saad wins or DIAA if Diaa wins.

Example

dimensions.in	standard output
2 2 2 2 3 2 5 3	DIAA SAAD

Problem E. Earnings

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

Three business men are sitting around a table. Each one of them has an amount of money. They know that making earnings can be difficult and time consuming so they decided to take a shortcut (not an ethical one though). They repeat the following step for a number of times:

- The amount of each business man becomes equal to the sum of the amounts of the other two.

They need your help to know the total money they have after a certain number of steps.

Input

The first line of input contains the number of test cases T ($T \leq 10$). The first line of each test case contains three integers A, B, C ($0 \leq A, B, C \leq 50$), the amounts of money that each business man starts with. The second line contains the number of steps N to perform ($0 \leq N \leq 10^9$).

Output

For each test case print the sum of the amounts that the business men have after repeating the process. The answer should be printed modulo $10^9 + 7$.

Example

earnings.in	standard output
2 0 1 0 2 2 3 7 5	4 384

Problem F. Coloring Balls

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

On a summer evening, Imane and Khalil were training together when Imane noticed Khalil's obsession with balls. Being a good friend, Imane gave Khalil a problem to solve on his way back home. Otherwise, his ride home would be pretty boring!

Given N balls each colored with some color, you are asked to count the number of ways to order these balls so that all balls of the same color are grouped together. Since the number of ways could be very large, you're asked to print the answer modulo $10^9 + 9$.

Khalil was too tired (and hungry) to solve this problem so he asked for your help. If you manage to solve this problem correctly, he might give you a bite of his sandwich!

Input

First line of input contains T ($1 \leq T \leq 100$), the number of test cases. Each of the following T lines contains N ($1 \leq N \leq 10^4$, the number of balls. The next line contains N balls, each having a color C ($1 \leq C \leq N$).

Output

For each test case, output a single line containing the answer to the problem modulo $10^9 + 9$.

Example

balls.in	standard output
3 5 3 4 2 1 5 1 1 3 2 1 1	120 1 4

Problem G. The Missing Digit

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

Each book is identified by an International Standard Book Number (ISBN). It was a 10-digit number but starting 2007 a new 13-digit number has been introduced.

However, till now, the old 10-digit number still be used and it will be considered for this problem.

The ISBN consists of 10 digits $d_1, d_2, \ldots, d_{10}$ such that:

- $0 \leq d_i \leq 9$ for $1 \leq i \leq 9$
- d_{10} is a check digit which is selected such that $(\sum_{i=1}^{10} i \cdot d_i)$ will be divisible by 11
- d_{10} could be any digit (from 0 to 9) or the uppercase letter "X" (used to represent 10)

Osama, an active student in the university, wants to buy some books online but he does not have a credit card. So, he printed the ISBNs for the required books and gave them to his friend Ahmad, who has a credit card, to buy them.

Unfortunately, there was a problem in Osama's printer; all ISBNs have been printed but with one missing digit per ISBN.

Could you help Ahmad know the missing digit of each ISBN.

Input

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

Each test case is specified on a single line which contains exactly 10 characters; only one of them is a question mark "?" specifying the position of the missing digit.

Output

For each test case, print the missing digit.

Example

missing.in	standard output
2 0071?44743 188?057584	2 9

Problem H. A Moroccan Citizen

Input file:	`citizen.in`
Output file:	`standard output`
Balloon Color:	`Purple`

A secret security agency has recruited you to make a specific task, because obviously you're the best coder in Morocco.

You'll be given a list of citizens' identifiers (ID) and you'll be asked multiple times whether or not a given ID belongs to a Moroccan citizen or not.

An ID is a string of 8 digits.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 10^5$) – the number of Moroccan citizens and the number of queries respectively.

The next N lines will contain N identifiers, each on a separate line.

The next M lines contain the query identifiers to verify.

Output

Print M lines, in each line print "YES"if the given ID is a Moroccan one, otherwise print "NO".

Examples

citizen.in	standard output
4 2 12458325 11789632 09874521 11475287 09452312 11475287	NO YES
1 2 11362988 09552785 11362988	NO YES

Problem I. Moving Digits

Input file:	`digits.in`
Output file:	`standard output`
Balloon Color:	`Gray`

You are given two boards a and b, each board has 1 row and n columns. Each cell in both boards can either be a digit from 0 to 9 or an empty cell represented by a dot character '`.`'.

You can perform the following operations on board a zero or more times:

- Move an odd digit $(1, 3, 5, 7, 9)$ one cell to the left if that cell is empty. For example, if $a =$ "`2..3`", you can make it "`2.3.`".
- Move an even digit $(0, 2, 4, 6, 8)$ one cell to the right if that cell is empty. For example, if $a =$ "`2..3`", you can make it "`.2.3`".

Your task is to check if the board a can be converted to the board b after applying as many operations as needed.

Input

The first line of the input contains a single integer T $(1 \leq T \leq 250)$, the number of test cases.

The first line of each test case contains a single integer n $(1 \leq n \leq 10^5)$, the number of columns of boards a and b.

The next two lines contain the boards a and b respectively. Each board is represented by a string of length n consisting of digits and dots.

The total sum of n over all test cases doesn't exceed 2×10^6.

Output

For each test case, print a single line with "`yes`" if board a can be converted to board b. Otherwise, print "`no`". The output is case **insensitive**.

Example

digits.in	standard output
4 3 2.3 23. 3 2.5 23. 9 012..3.1. 01..231.. 9 012..301. 01..23..1	yes no yes no

Problem J. Sorted Array Partitioning

Input file:	`sorted.in`
Output file:	`standard output`
Balloon Color:	`Gold`

You are given a **sorted** array of n integers, and a number k that represents the number of partitions you must split the array into.

The elements of one partition must form a contiguous part of the array. Each element in the array must belong to exactly one partition.

For each partition from left to right, you are given one information about it: value x appears exactly y times in this partition.

Find if it is possible to split the array into k partitions, such that each partition satisfies the given information about it.

Input

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

The first line of each test case contains two integers n and k ($1 \leq k \leq n \leq 10^5$), the number of elements in the array and the number of partitions, respectively.

The second line contains n sorted integers a_1, a_2, ..., a_n ($1 \leq a_i \leq n$), the element of the array. It is guaranteed that ($a_{i-1} \leq a_i$) for all i ($i > 1$).

Each of the following k lines contains two integers x_i and y_i ($1 \leq x_i, y_i \leq n$), representing that the value x_i must appear y_i times in the i^{th} partition from left.

The total sum of n over all test cases doesn't exceed 10^6.

Output

For each test case, print a single line containing "`yes`" if it is possible to partition the array. Otherwise, print "`no`". The output is case **insensitive**.

Example

sorted.in	standard output
3 9 3 1 1 1 3 3 4 7 7 7 1 2 3 2 7 2 5 2 3 3 5 5 5 5 3 3 2 4 2 1 1 2 2 1 1 2 3	yes no no

Note

In the first test case, we can split the array in this way: 1 1 | 1 3 3 4 7 | 7 7, the first partition has two 1s, the second partition has two 3s, and the last partition has two 7s, as required.

Problem K. Constraints

Input file:	`product.in`
Output file:	`standard output`
Balloon Color:	`Silver`

You are given an array a of n positive integers, and m sub-array constraints that must be satisfied.

A subarray of the array a is a sequence of consecutive elements of it: a_l, a_{l+1}, ..., a_r, for some integers (l, r) such that $1 \leq l \leq r \leq n$.

Each constraint requires the product of a subarray to have a specific parity (either even or odd).

Your task is to change the minimum number of elements in the array a so that all the constraints are satisfied, if that is possible.

Input

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

The first line of each test case contains two integers n and m ($1 \leq n, m \leq 10^5$), the size of the array a and the number of constraints, respectively.

The second line contains n integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq 10^9$), the elements of the array.

Each of the following m lines contains three integers l, r, and p ($1 \leq l \leq r \leq n$, $p \in 0, 1$), representing a constraint on the subarray $[l, r]$. If p is 0, the product of the subarray elements $a_l \times a_{l+1} \times \cdots \times a_r$ must be even, otherwise it must be odd.

The total sums of n and m over all test cases do not exceed 2×10^6.

Output

For each test case, on a single line, if it is impossible to satisfy all the constraints, print -1. Otherwise, print the minimum number of elements that need to be changed.

Example

product.in	standard output
2 5 3 3 7 2 5 3 3 5 0 1 3 1 2 4 0 3 3 1 2 3 1 2 1 2 3 1 2 2 0	2 -1

Problem L. Cutting Swiss Cheese

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

You have a dinner party, and you hope k people will be there. For the occasion, you've bought a circle of cheese that you want to cut into equal slices, one for each guest. However, you have bought Swiss cheese which has many bubbles in it. The cheese is represented as a two-dimensional circle of radius R with many other circles (bubbles) strictly inside it. No bubble will touch the outer border or the center of the cheese circle.

After cutting the slices, you will remove all the crumbs; crumbs are parts that are disconnected from the center of the cheese circle, no matter how large they are.

Find if it's possible to cut the cheese into slices such that after removing the crumbs, the outer border of the cheese remains as a complete circle of radius R. Since you are not sure if all your friends will come, find the answer for all number of slices from 1 to k. Note that the entire cheese should be divided, resulting in the required number of slices, no less or more.

Please check the picture and the notes in the following page.

Input

The first line of input contains a single integer T ($1 \leq T \leq 16$), the number of test cases.

The first line of each test case contains three integers n, R, and k ($1 \leq n \leq 2000$, $3 \leq R \leq 10^5$, $1 \leq k \leq 100$), the number of bubbles in the cheese, the radius of the cheese circle, and the maximum number of slices you may need.

The i^{th} of the following n lines will contain three integers x_i, y_i and r_i, the x and y coordinates of the i^{th} bubble and its radius, respectively.

It's guaranteed that all bubbles are strictly inside the cheese circle and no bubble will touch or cover the center of the cheese circle.

Output

For each test case, print a single line with k characters, the i^{th} character should be 1 if it is possible to cut the cheese into i slices according to the statement, otherwise it should be 0.

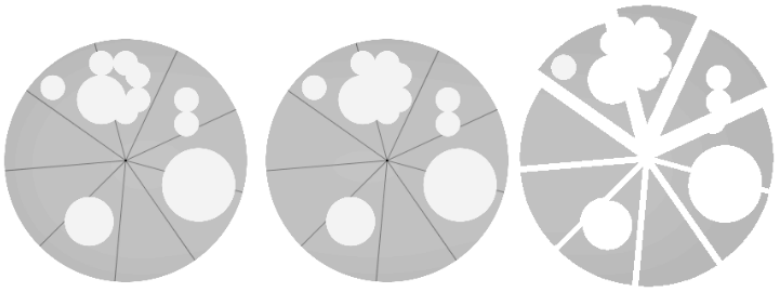
It's guaranteed that the answer will not change if we increase or decrease the radius of all bubbles by 10^{-6}.

Example

cheese.in	standard output
2 11 10 10 1 7 1 0 8 1 1 5 1 0 4 1 5 3 1 -2 8 1 6 -2 3 5 5 1 -3 -5 2 -2 5 2 -6 6 1 1 10 17 4 -4 3	1111111110 11111111111000000

Note

For the first test case, it is possible to find a way to cut the cheese to any numbers of slices from 1 to 9, but not 10. The picture below shows one way to cut the cheese into 9 slices.



You can see the state after removing the crumbs in the middle, and to the right we have exactly 9 slices, and the outer part of each slice is an arc of the cheese circle.

Problem M. Flow Sources

Input file:	`flow.in`
Output file:	`standard output`
Balloon Color:	`Aqua`

You are given an undirected tree of size n, where each edge has one of the labels: up, down, left, or right.

The tree represents a water piping system. Initially, the pipes are dry and we will choose **one** source node and start pumping water to it.

When water reaches a node through some edge, it will be pushed through all other edges. So for each edge, the water will flow in one direction only.

For a given source, the nodes must be positioned such that the edges will be oriented relative to the direction of the water. For example, if water flows from node a to node b through an edge with label "`right`", then node b must be to the right of node a.

You can imagine it as drawing the tree on a 2D plane, first you place the source node, then place its adjacent nodes relative to it according to the labels of the edges, and so on. Nodes and edges can intersect and overlap.

The only constraint is that no node can have overlapping edges moving water from or to it, which means we need to choose the source such that the following two cases will not happen:

- Water flows from some node through two or more edges of the same label.
- Water flows from a to b through an edge of some label, and flows from b to c through an edge of the opposite label (up and down, or left and right). This is invalid because we have two overlapping edges moving water from or to b.

Find the possible sources where we can start pumping water without having overlapping edges moving water from or to the same node. It is guaranteed that the given tree will have at least one possible source.

Input

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

The first line of each test case contains a single integer n ($2 \leq n \leq 10^5$), the number of nodes in the tree.

Each of the following $n-1$ lines contains two integers u and v ($1 \leq u, v \leq n$), followed by one letter c ($c \in$ {U, D, L, R}), representing an edge between the two nodes u and v with label c. The four letters U, D, L, R represent the directions up, down, left, and right, respectively.

It is guaranteed that the given graph is a tree, and it has at least one possible source where we can start pumping water.

The total sum of n over all test cases doesn't exceed 10^6.

Output

For each test case, print the possible sources in increasing order, separated by at least one space, on a single line.

Example

flow.in	standard output
4 5 1 2 R 2 3 R 3 4 R 4 5 R 8 1 2 L 4 1 U 1 5 D 1 3 L 2 6 L 8 6 D 7 6 U 2 1 2 U 4 3 2 L 2 4 L 1 3 R	1 5 3 6 1 2 3

Note

In the first test case, we can start at node 1 or node 5, in both cases we won't have overlapping edges of the same node. If we start at any other node, we will have two edges with label "R" going from the same node.

Problem N. Penalty

Input file:	`penalty.in`
Output file:	`standard output`
Balloon Color:	`LimeGreen`

Ahmed is contesting in the MCPC 2020 edition. One of the problems is to compute the penalty of a submission knowing when it got accepted the first time, can you help him do that. (Yes, You are his teammate, so it's not considered cheating).

The submissions's penalty is computed as the number of minutes to get the first accepted plus an additional 20 min for every wrong submission.

Input

First line will contain the number of test cases $1 \leq T \leq 100$.

T lines follow, each contain the two integers $1 \leq P, N \leq 300$ The total number of submissions (including the one that got accepted), and the first time the problem got accepted.

Output

Print T lines each containing the answer for every test case.

Example

penalty.in	standard output
3 2 10 1 12 200 200	30 12 4180