Problem A. YouCube

Program:	`expressions.(c|cpp|java|py)`
Input:	`youcube.in`
Balloon Color:	`Black`

YouCube is a website that has M users and N videos and each video has a unique ID in the range $[1, N]$ and a duration of X minutes.

Each user has a favorite play list that consists of K videos.

The following are some of the website features:

1. Once a user logs in, the website suggests a video for that user to start watching.
2. Once a user finishes watching a video, the website suggests another video for him to watch next.

A User gets bored in one of the following scenarios, and once any of them occurs he will close the website immediately:

1. Watched more than W videos
2. Watched a video that does not exist in his play list
3. Watched the same video more than once.

Given Youcube's videos list with duration and each user's play list, we want to suggest a video for each user to watch next, in such a way that maximizes the total duration of all the watched videos of all users, print that maximal possible value (the maximum total duration watched by all users).

Input

The first line will contain the number of test cases $1 \leq T \leq 50$ then T test cases follow.

Each test case will consist of the following:

The first line consists of three integers N, M and W ($1 \leq N, M, W \leq 10^3$), the number of videos that exist on the website, the number of users and the maximum number of videos a single user can watch respectively.

Then M lines will follow, the i^{th} line represents the details of the i^{th} user's play list. Each line will start with an integer K_i ($0 \leq K_i \leq N$) (the number of videos in the i^{th} user's play list) followed by K_i distinct integers $ID_{i,1}$, $ID_{i,2}$, ..., ID_{i,K_i} ($1 \leq ID_{i,j} \leq N$) representing the IDs of the videos in the play list of the i^{th} user.

The last line of each test case will contain N integers X_1, X_2, ..., X_N ($1 \leq X_i \leq 10^7$) represent the duration of all the videos that exist in the website (i.e the duration of the video with an $ID = i$ is X_i).

Output

For each test case, print a single line which contains the maximum duration sum of all videos users will spend watching on the website, considering that suggestions are done optimally.

Example

youcube.in	Standard Output
3 4 2 3 2 1 2 1 1 21 17 24 16 3 3 1 2 1 2 2 1 2 2 2 3 26 22 26 5 5 2 2 1 2 3 1 2 3 3 1 2 5 4 1 2 3 5 2 2 3 5 3 9 5 10	59 78 68

Note

This problem has a large input file, Use fast Input and output

Problem B. Nearest Tens

Program:	`youcube.(c|cpp|java|py)`
Input:	`tens.in`
Balloon Color:	`Pink`

Given a positive integer N, round it to the nearest tens.

Input

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

Then T lines follow, each contains a single integer N (fits in 32-bit signed integer).

Output

For each test case output a single integer, which represents N rounded to the nearest tens.

Example

tens.in	Standard Output
5 34 37 28 16 5	30 40 30 20 10

Note

- If the number you are rounding is followed by 5, 6, 7, 8, or 9, round the number up. Example: 38 rounded to the nearest ten is 40.
- If the number you are rounding is followed by 0, 1, 2, 3, or 4, round the number down. Example: 33 rounded to the nearest ten is 30.

Problem C. Unpalindromize

Program:	`revive.(c|cpp|java|py)`
Input:	`unpalindromize.in`
Balloon Color:	`Brown`

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

Given a String S. Print the minimum number of moves needed to make this string **not palindrome**.

The only allowed move is to swap two **adjacent** characters in the string.

Input

The first line will contain number of test cases T.

Each test case will consist of a string S ($1 \leq |S| \leq 10^5$) contains lowercase English letters.

It is guaranteed that the total length of all strings won't exceed 10^6.

Output

For each test case print one line containing minimum number of moves needed to make this string not palindrome or print -1 if it is impossible to make the string not palindrome by any number of moves

Example

unpalindromize.in	Standard Output
3 rbbbbr fffff rcbyn	1 -1 0

Problem D. Revive Tree

Program:	`simple.(c|cpp|java|py)`
Input:	`revive.in`
Balloon Color:	`Light green`

A tree is a graph that has a node called the root and all its nodes have exactly one in-edge but the root.

A DFS (Depth First Search) could be applied to a tree starting from the root and generating an array of the nodes order called the DFS order array by the following pseudo code:

– $index = 0$
– $function\ dfs(node)$
—- $a[++index] = node$
—- $for\ each\ child\ of\ node$
—- $dfs(child)$

Given a DFS order array a of a rooted **directed** tree and the sub-tree size array b that contains a number a_i represents the count of nodes in the sub tree rooted by node i, you are to print the edges of the original tree or print -1 if there is no such tree.

Input

The first line contains the number of test cases ($1 \leq T \leq 100$), then for each test case: The first line contains ($1 \leq N \leq 100000$) which is the number of nodes in the tree. The second line contains N numbers, the dfs order array ($1 \leq a_i \leq N$). The last line contains N numbers, the subtree size of each node ($1 \leq b_i \leq N$).

Output

Print -1 if there is no such tree. Otherwise, print $N - 1$ lines, the edges of the tree in any order.

Example

revive.in	Standard Output
4 4 1 2 3 4 4 3 2 1 6 1 2 5 6 4 3 6 4 1 1 1 1 4 2 4 1 3 1 4 1 3 3 1 2 3 1 2 3	1 2 2 3 3 4 1 2 2 5 2 6 2 4 1 3 2 4 4 1 4 3 -1

Note

This problem has a large input file, use fast input and output

Problem E. Simple Task

Program:	`time.(c|cpp|java|py)`
Input:	`simple.in`
Balloon Color:	`Silver`

Given two integers a and b. Starting from a, your goal is to reach b.

You can only use the following two operations:

1. Add $+1$ to a.
2. Multiply a by -1.

Your task is to find the minimum number of operations required to reach b.

Input

The first line will contain T ($1 \leq T \leq 1000$) the number of test cases.

Each test case contains two integers a and b ($-10^9 \leq x \leq 10^9$).

Output

For each test case print the minimum number of operations required to reach b.

Example

simple.in	Standard Output
3 10 20 10 -10 -10 -20	10 1 12

Problem F. Congratulations

Program:	`tree.(c|cpp|java)`
Input:	`acpc.in`
Balloon Color:	`Red`

Congratulations!!! your team have qualified to the 2018 ACPC (Arab Collegiate Programming Contest) — You made it!

Everyone is waiting for that moment, when his team name is announced from the qualified teams to the 2018 ACPC that will be held in Sharm El Sheikh.

Your team now is qualified and the contest is about to begin in X minutes and your team needs Y minutes to reach the contest hall. Write a program that will tell you if you can reach in time or not.

Input

First line will contain only one integer T $(1 \leq T \leq 100)$ − the number of test cases.

Each test case will contain two numbers X and Y $(0 \leq X, Y \leq 10^6)$ − the time that the contest is about to begin and the time needed to reach the contest hall.

Output

If your team can reach the contest in time print "YES otherwise print "NO".

Example

acpc.in	Standard Output
2 15 23 89 23	NO YES

Problem G. Pyramid Sequence

Program:	`unpalindromize.(c|cpp|java)`
Input:	`pyramid.in`
Balloon Color:	`Orange`

A pyramid sequence is a sequence that have some (possibly zero) non-decreasing integers then some (possibly zero) non-increasing integers. for example: $\{1, 2, 3, 4, 4, 3, 2\}, \{1, 2, 2\} and \{3, 2, 1\}$ are valid pyramids, but $\{1, 2, 1, 2\} and \{3, 1, 2\}$ are not.

Each pyramid sequence has a score equals the difference between the max and the min integers in that sequence.

Given a pyramid sequence of size N, count how many sub-arrays of the given sequence have the same score as the whole sequence.

More formally if the score of the whole sequence is X, then for each L and R ($1 \leq L \leq R \leq N$) how many times we will have Score of this sub-array $[L, R] = X$.

Input

The first line will contain an integer $1 \leq T \leq 100$ the number of test cases.

Each test case contains two lines:
the first line has one integer N, the size of the pyramid sequence ($1 \leq N \leq 10^5$)
the second line has N integers, the values of the pyramid ($1 \leq A_i \leq 10^9$).

it's guaranteed that this sequence will have some (possibly zero) non-decreasing integers then some (possibly zero) non-increasing integers.

Output

For each test case you print the number of sub-arrays that have the same score as the whole sequence.

Example

pyramid.in	Standard Output
2 7 1 2 3 4 4 3 2 5 1 1 2 2 3	4 2

Note

This problem has a large input file, use fast input and output

Problem H. Twins

Program:	`pyramid.(c|cpp|java)`
Input:	`twins.in`
Balloon Color:	`Yellow`

Sabek and Lahek are twin brothers, they love playing with race cars.

Each day one of them go buy an item for his own car.

Given each day report containing who bought and how many items were bought, your task is to tell after each day whether if they have the same amount of items.

Input

The first line will contain ($1 \leq T \leq 10$) the number of test cases.

Each test case contains one integer d, the number of days ($1 \leq d \leq 10^5$)
after that d lines following each line will contain 2 integers, the first one has value of "1"for (Sabek) or "2"for (Lahek), and the second one is the number of the items bought ($0 \leq x \leq 10^9$)

Output

for each test case you will print d lines containing "YES"if both Sabek and Lahek have the same amount of items after that day, and "NO"otherwise.

Example

twins.in	Standard Output
1 6 1 1 2 1 2 3 2 5 1 5 1 3	NO YES NO NO NO YES

Problem I. Balanced Tree

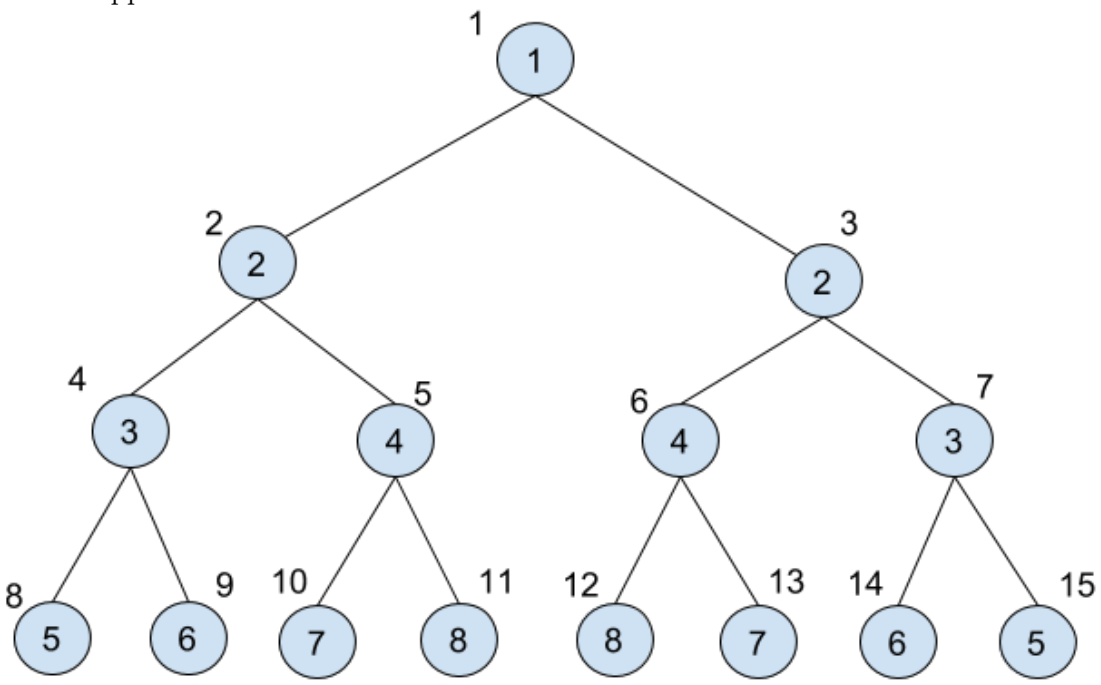
Program:	`twins.(c|cpp|java)`
Input:	`tree.in`
Balloon Color:	`Purple`

A binary tree is a tree that starts from one node called the root and each node has at most two children but the last nodes (the leafs), they have no children.

Given a binary tree of n nodes numbered from 1 to n, each node have a value a_i such that $1 \leq i \leq n$. We want to check if the tree is oppositely balanced where an oppositely balanced tree has the following properties:

1. The tree must be a full binary tree (a binary tree whose nodes have exactly two children but the leafs have no children).
2. Each sub-tree must has the same sum of its opposite sub-tree; the sub-tree sum of a node is defined as the sum of all the node values formed by the sub-tree rooted at that node (including the node itself).

The following image illustrates how a sub-tree is opposite to another one, that each two sub-trees with the same value are opposite to each others:



In other words, we could see the tree rooted by node 1 to have its two sub trees rooted by its children (2 and 3) to be mirrored.

The tree will be given as an array of n integers $a_1, a_2, ..., a_n$, that a_1 is the root's value, and for each node i, its left node will be in $a_{(2*i)}$ and the right node will be in $a_{(2*i+1)}$.

Input

The first line will contain number of test cases $1 \leq T \leq 100$. Each test case will consists of two lines:

The first line will contain only one integer n $(1 \leq n \leq 10^5)$ $--$ the number of nodes in the tree.

The second line will contain n integers $a_1, a_2, ..., a_n$ $(1 \leq a_i \leq 10^{18})$ $--$ the values of each node in the tree.

Output

For each test case, print "YES"if the given tree is balanced, otherwise print "NO".

Example

tree.in	Standard Output
3 7 5 1 1 2 3 3 2 6 5 1 1 2 3 3 7 5 5 1 2 3 3 2	YES NO NO

Note

This problem has a large input file, use fast input and output

Problem J. A String

Program:	`acpc.(c|cpp|java)`
Input:	`string.in`
Balloon Color:	`White`

We have a string X, find another string Y that contains exactly the same characters of X in any order. (note that the strings are case sensitive).

Input

The first line will contain the number of test cases T.

Then T lines follow, each consists of a single alphabetic string X $(1 \leq |X| \leq 100)$.

Output

For each test case, output, in a single line, a string Y that contains exactly the same characters of the input string in any order (case sensitive).

If there are multiple answers, print any one of them.

Example

string.in	Standard Output
4 uBiPhTqY afmEuh SuoUM XcxtWIeoE	hBqiuTYP umafEh oMSuU EWeIctoxX

Problem K. Short On Time!

Program:	`tens.(c|cpp|java)`
Input:	`time.in`
Balloon Color:	`Green`

Hamada was going out with his friends to the Odd Canteen Pizzeria Chophouse (OCPC), he was busy preparing the ultimate problem for you to solve, so he totally forgot about his friends.

Now Hamada doesn't have enough time, and he wants to catch up with his friends in the OCPC.

There are m roads that he should pass to get there and the i^{th} road has a_i parts.

The j^{th} part has its length len_j, and the car maximum speed when passing in it cannot become larger than v_j.

Now Hamada is wondering, how long will it take him to pass all the roads, can you help him?

Input

The first line of the input contains one integer t ($1 \leq t \leq 200$) , the number of test cases.

The first line of the each test case contains one integer m ($1 \leq m \leq 100$).

Followed by m blocks, the i^{th} block contains one integer a_i ($1 \leq a_i \leq 100$) the number of parts.

Followed by a_i lines, the j^{th} line has 2 integers len_j ($1 \leq len_j \leq 100$) and v_j ($1 \leq v_j \leq 100$), the length and the maximum velocity that describe the j^th part.

Output

For each test case, print one real number, the time it will take Hamada to pass all the roads.

Your output will be considered correct if the absolute or relative error $<= 10^{-6}$.

Example

time.in	Standard Output
1 2 2 6 2 3 3 2 6 3 1 1	7.000000

Note

In the sample there are 2 roads, each has 2 parts, Hamada will spend $(3 + 1) = 4$ unit of times in first road, and $(2 + 1) = 3$ units of time in the second road, resulting in 7 units of time in total.

Problem L. Expressions

Program:	`string.(c|cpp|java)`
Input:	`expressions.in`
Balloon Color:	`Blue`

This is Alice's first day of school after a long vacation.
The first class is Mathematics and she is currently learning about expressions, simply; she is given an expression in the following form:

- $a_1X^{p_1}+a_2X^{p_2}+a_3X^{p_3}+...+a_{n-1}X^{p_{n-1}}+a_nX^{p_n}$

Where a_i is the coefficient of term i, p_i is the power in term i and n is the number of terms in the expression.
In other words, the i^{th} term of the expression is represented as $a_i * X^{p_i}$.
(i.e if $a_i = 4$, $p_i = 2$ and $X = 3$ then term i should be calculated as $4 * (2^3) = 32$)

Given an expression in the above described form and the value of the variable X, Alice wants to calculate the final result for the formula.

Input

The first line will contain number of test cases T ($1 \leq T \leq 10^3$) then T test cases follow.
Each test case will consists of two lines:
The first line contains a string S ($3 \leq |S| \leq 60$) that represents the expression in the described form where for each i ($1 \leq i \leq n$), ($1 \leq a_i, p_i \leq 9$). It is guaranteed that the given expression is in the correct form.

The second line will contain an integer X ($1 \leq X \leq 9$) – the value of the variable X in the expression.

Output

For each test case, print a single line containing only one integer represents the final result of the expression.

Example

expressions.in	Standard Output
3 1X1 3 2X1+4X5+3X1 2 5X3+3X2+5X2+4X1 4	3 138 464