Problem A. Future of Tencent

Kelly is now in a busy team which has a lot of interesting and challenging projects. His boss wants to hire more excellent software engineers.

So in this problem, your task is to recommend your excellent friends to Kelly.

If you have recommended at least one friend to Kelly, he will be so happy to praise you: “You are the future of Tencent!”; Otherwise, you will get a wish from him: “Good luck and Enjoy TPC!”;

Given the number of friends you recommended to Kelly, do you know what Kelly will say to you?

Input

There are multiple test cases. The first line of the input contains an integer T (about 200) indicating the number of test cases. For each test case:

The first and only line contains one integer n ($0 \leq n \leq 100$) indicating the number of friends you recommended to Kelly.

Output

For each test case output one line containing one string indicating what Kelly will say (without quotes).

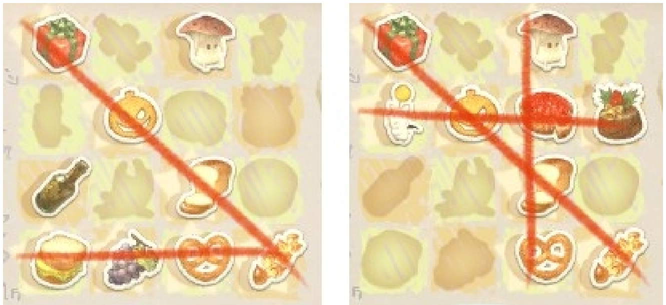
Example

standard input	standard output
3 1 0 20	You are the future of Tencent! Good luck and Enjoy TPC! You are the future of Tencent!

Problem B. Wondrous Tails

Wondrous Tails is a weekly activity of the famous MMORPG *Final Fantasy XIV* that allows players to complete old content to earn rewards. Each week the player will receive a journal containing some objectives and a grid of 4 rows and 4 columns. When the player completes an objective, a seal will be stamped into a random empty cell of the grid. The player will be rewarded according to the pattern of the seals when he collects 9 seals in the journal.

To be more precise, for each row/column/diagonal of the grid (by the diagonals of the grid we mean both the main diagonal and the anti diagonal), if all its 4 cells are filled with a seal we say this row/column/diagonal forms a line. For top level award a grid containing at least 3 lines is needed.



A grid with 2 lines (left) and a grid with 3 lines (right)

It's obvious that for a randomly generated seal pattern it's almost impossible to receive the top award, so the players are allowed to rearrange the pattern in some way. For the simplicity of this problem let's assume that the player can perform the following operation on a grid with 9 seals any number of times: select one seal and move it to any empty cell.

Given a grid with 9 seals, what's the minimum number of operations one need to perform to change it into a grid with at least 3 lines?

Input

There are multiple test cases. The first line of the input contains an integer T (about 10^4) indicating the number of test cases. For each test case:

The only four lines each contains a string indicating the given grid. Let s_i be the string on the i-th line ($s_{i,j} \in \{$'.' (ASCII: 46), '#' (ASCII: 35)$\}$, $|s_i| = 4$). If $s_{i,j}$ = '.' then the cell on the i-th row and j-th column is an empty cell, while $s_{i,j}$ = '#' indicates that the cell on the i-th row and j-th column contains a seal. It's guaranteed that there are exactly 9 '#'s in the given grid.

Output

For each test case output one line containing one integer, indicating the minimum number of operations needed to change the given grid into a grid with at least 3 lines.

Example

standard input	standard output
2 #### #### #... .... .#.# #### .#.. ##..	2 0

Problem C. BFS Sequence

Given two sequences $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ where both sequences are permutations of $1, 2, \ldots, n$, please find a rooted tree such that both sequences are the BFS (breadth first search) sequence of the tree, and the height of the tree is as large as possible.

Recall that

- The height of the tree is the maximum number of edges between the root of the tree and any vertex on the tree. That is to say, a tree with only one vertex has a height of 0, and a tree with two vertices has a height of 1.
- A BFS sequence of a rooted tree can be produced as follows: Create a queue q containing only the root of the tree, then repeatedly remove h, that is the head of q, and add all children of h to the tail of q in arbitrary order until q is empty. Arrange the vertices by the order they are removed and we get a BFS sequence of the tree.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line of the input contains an integer n ($1 \le n \le 10^5$) indicating the length of the BFS sequences.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) which is a permutation of n indicating the first BFS sequence.

The third line contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le n$) which is also a permutation of n indicating the second BFS sequence.

It's guaranteed that the sum of n of all test cases will not exceed 10^6.

Output

For each test case, if there exists a valid answer first output "Yes" (without quotes) in one line, then in the next line output n integers $p_1, p_2, \ldots, p_n$ separated by a space where p_i indicates the parent of vertex i ($p_i = 0$ indicates that vertex i is the root of the tree). If there is no valid answer just output "No" (without quotes) in one line.

If there are multiple valid answers you can output any of them.

Please, DO NOT output extra spaces at the end of each line, or your solution may be considered incorrect!

Example

standard input	standard output
2 8 3 4 8 1 5 6 7 2 3 1 8 4 2 7 5 6 2 1 2 2 1	Yes 3 1 0 3 4 4 8 3 No

Problem D. Persistent String

Given a string s, we're going to perform q operations of the following three types on the string:

- $+\ p_i\ t_i$: Insert a string t_i into s after the p_i-th character;
- $!\ a_i\ b_i\ c_i\ p_i$: Make c_i copies of substring of s starting from the a_i-th character and ending at the b_i-th character (both inclusive), concatenate them, and insert it into s after the p_i-th character;
- $?\ p_i$: Query the p_i-th character of s.

Note that all strings use 1-based index and $p_i = 0$ indicates that we're making insertions at the beginning of s.

Your task is to calculate the results for the third type of operations.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains a string s ($1 \le |s| \le 10^5$) indicating the initial string.

The second line contains an integer q ($1 \le q \le 5 \times 10^3$) indicating the number of operations.

Each of the next q lines contains the description of each operation. Let's denote L_i as the length of s before the i-th operation. The constraints for each type of operations are as follows:

- $+\ p_i\ t_i$: $0 \le p_i \le L_i \le 10^{18}$, $1 \le |t_i| \le 10^6$;
- $!\ a_i\ b_i\ c_i\ p_i$: $0 \le p_i \le L_i \le 10^{18}$, $1 \le a_i \le b_i \le L_i$, $1 \le c_i \le 10^{18}$;
- $?\ p_i$: $1 \le p_i \le L_i \le 10^{18}$.

It is guaranteed that all strings consist only of lowercase English letters. It is also guaranteed that the sum of q in all test cases does not exceed 5×10^3 and the sum of $|s|$ and $|t_i|$ in all test cases does not exceed 10^6.

Output

For each operation of the third type output one line containing a character denoting the answer.

Example

standard input	standard output
1 abcdedgh 17 ? 5 ! 3 5 6 7 ? 7 ? 8 ? 9 ? 10 ? 26 + 0 uvw ? 1 ? 2 ? 3 + 13 xyz ? 13 ? 14 ? 15 ? 16 ? 17	e g c d e h u v w e x y z c

Problem E. Minimum Spanning Tree

Given an undirected connected graph with n vertices and m edges, where the i-th edge is associated with two parameters a_i and b_i.

Let $f(x)$ be the sum of weights of the edges in the minimum spanning tree when the weight of the i-th edge is $a_i + b_i \cdot x$, your task is to calculate $\min(f(l), f(l+1), \ldots, f(r))$.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains four integers n, m, l and r ($2 \le n \le 10^5$, $n - 1 \le m \le 2 \times 10^5$, $0 \le l \le r \le 10^6$), indicating the number of vertices, the number edges and the range of x.

For the next m lines, the i-th line contains four integers u_i, v_i, a_i and b_i ($1 \le u_i, v_i \le n$, $u_i = v_i$, $1 \le a_i \le 10^6$, $-10^6 \le b_i \le 10^6$), indicating that the i-th edge connects vertices u_i and v_i and the parameters of the i-th edge. It is guaranteed that the graph is connected.

It is guaranteed that the sum of n in all test cases will not exceed 10^5 and the sum of m in all test cases will not exceed 2×10^5.

Output

For each test case output one line containing one integer indicating the answer.

Example

standard input	standard output
2 5 6 1 5 1 2 3 1 2 3 5 -1 3 4 1 2 4 5 1 -1 5 1 5 3 2 4 3 -1 5 6 1 5 1 2 1 1 2 3 1 2 3 4 1 -10 3 4 2 10 5 1 3 10 2 4 5 -10	2 -35

Problem Tutorial: "Future of Tencent"

按题意输出即可。

Problem Tutorial: "Wondrous Tails"

实际上只有 24 种 pattern 可以达成三线，且一定是横着一条线，竖着一条线，对角线一条线。设横着的线在第 i 行，竖着的线在第 j 列。若主对角线形成一条线则 $i \neq j$ 时都能达成三线；若副对角线形成一条线则 $i + j \neq 5$ 时都能达成三线。枚举 24 种 pattern 并选出最优解即可。

Problem Tutorial: "BFS Sequence"

首先检查 a_1 与 b_1 是否相等，不相等则无解，否则 a_1 就是树的根。

设 h_i 表示点 i 与根之间边的数量。若 $h_i < h_j$ 则点 i 在任意 BFS 序列里一定排在点 j 的前面，若 $h_i = h_j$ 则点 i 和点 j 之间的顺序不一定。也就是说，BFS 序列一定可以按节点的 h 值分成高度加 1 段。

那么问题转化为：将两个序列在相同位置划分成尽量多段，使得两个序列对应的段中含有相同元素。分段数量减 1 就是答案的上界。

接下来构造符合这个上界的答案。我们只考虑序列 $a_1, a_2, , a_n$，对于相邻的两端，只需要把后一段中的所有节点全部作为前一段任意一个节点的子节点即可。

Problem Tutorial: "Persistent String"

显然我们可以用可持久化平衡树来维护所有操作，但这个不免有点复杂。观察到 n 比较小，最大只有 5000，查询的时候可以考虑倒着暴力模拟所有插入操作，然后找到这个字符来自 s 或者 t_i 的哪个位置。

就是说，我们要查询当前 S_i 的第 p_i 个字符，那么需要求出这个字符在 S_{i-1} 的对应位置 p_{i-1}，其中 S_i 表示第 i 次操作前的字符串。

- 如果第 $i-1$ 个操作是 +，那么这次操作在 S_i 中对应的区间是 $[pos_{i-1}, pos_{i-1} + |t_{i-1}|)$，如果在区间左侧，$p_{i-1}$ 和 p_i 一样；如果 p_i 在这个区间里，那么就确定了这个字符；如果在区间右侧，那么 $p_{i-1} = p_i - |t_{i-1}|$。
- 如果第$i-1$个操作是 ?，那么这次操作在 S_i 中对应区间为 $[pos_{i-1}, pos_{i-1} + (b_{i-1} - a_{i-1} + 1) \cdot c_{i-1})$。 类似的， 如果在区间左侧，$p_{i-1} = p_i$； 如果在区间里，$p_{i-1} = a_{i-1} + (p_i - pos_{i-1}) \bmod (b_{i-1} - a_{i-1} + 1)$；如果在区间右侧，$p_{i-1} = p_i - (b_{i-1} - a_{i-1} + 1) \cdot c_{i-1}$。

Problem Tutorial: "Minimum Spanning Tree"

由于每条边的权值是一个关于 x 的一次函数，所以一棵生成树的权值也是关于 x 的一次函数。所以这一题实际上就是求很多一次函数的最小值。

我们知道，求许多一次函数的最小值其实就是维护一个上凸壳。那么整个上凸壳的最小值一定在两端取到。因此，我们只需要求出 $x = l$ 或者 $x = r$ 时候的最小生成树的权值，两者取最小即可。