Problem A. Yes, Prime Minister

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	512 mebibytes

Mr. Hacker's Department of Administrative Affairs (DAA) has an infinite amount of civil servants. Every integer is used as an ID (identification number) by exactly one civil servant. Mr. Hacker is keen on reducing overmanning in civil service, so he will only keep people with consecutive IDs in $[l, r]$, and dismiss all the others.

However, permanent secretary Sir Humphrey's ID is x, and he cannot be kicked out, so $l \le x \le r$ must hold. Mr. Hacker wants to be Prime Minister, so he demands that the sum of people's IDs $\sum_{i=l}^{r} i$ must be a prime number.

You, Bernard, need to make the reduction plan which meets the demands of both bosses. Otherwise, Mr. Hacker or Sir Humphrey will fire you.

Mr. Hacker would be happy to keep as few people as possible. Please calculate the minimum number of people left to meet their requirements.

A prime number p is an integer greater than 1 that has no positive integer divisors other than 1 and p.

Input

The first line contains an integer T ($1 \le T \le 10^6$), the number of test cases. Then T test cases follow.

The first and only line of each test case contains one integer x_i ($-10^7 \le x_i \le 10^7$), the ID of Sir Humphrey.

Output

For each test case, print a line with one integer: the minimal number of people kept if such a plan exists, or -1 otherwise.

Example

standard input	standard output
10 -2 -1 0 1 2 3 4 5 6 7	6 4 3 2 1 1 2 1 2 1

Problem B. Might and Magic

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Two heroes are fighting, whose names are hero 0 and hero 1 respectively.

You are controlling the hero 0, and your enemy is the hero 1. Each hero has five integer attributes: ATTACK, DEFENSE, POWER, KNOWLEDGE, and HEALTH. When two heroes battle with each other, they will take turns to attack, and your hero moves first. One hero can make **exactly one attack** in one turn, either a physical attack or a magical attack.

Assume their attributes are A_i, D_i, P_i, K_i, H_i ($0 \le i \le 1$). For hero i, its physical attack's damage is $C_p \cdot \max(1, A_i - D_{1-i})$, while its magical attack's damage is $C_m \cdot P_i$. Here, C_p and C_m are given constants.

After hero i's attack, H_{1-i} will decrease by the damage of its enemy. If H_{1-i} is lower or equal to 0, the hero $(1-i)$ loses, the hero i wins, and the battle ends.

Hero i can make magical attacks no more than K_i times in the whole battle.

Now you know your enemy is a Yog who is utterly ignorant of magic, which means $P_1 = K_1 = 0$, and he will only make physical attacks. You can distribute N attribute points into four attributes A_0, D_0, P_0, K_0 arbitrarily, which means these attributes can be any non-negative integers satisfying $0 \le A_0 + D_0 + P_0 + K_0 \le N$.

Given C_p, C_m, H_0, A_1, D_1, and N, please calculate the maximum H_1 such that you can build hero 0 and fight so that it wins the game.

Input

The first line contains an integer T ($1 \le T \le 10^4$), the number of test cases. Then T test cases follow.

The first and only line of each test case contains six integers C_p, C_m, H_0, A_1, D_1, N ($1 \le C_p, C_m, H_0, A_1, D_1, N \le 10^6$), the attributes described above.

Output

For each test case, print a line with one integer: the maximum enemy health such that it is possible to win.

Example

standard input	standard output
2 1 1 4 5 1 4 2 5 1 9 9 6	4 25

Problem C. 0 Tree

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

We have a tree $\langle V, E\rangle$ that consists of n vertices numbered from 1 to n. Each vertex $i \in V$ has weight a_i. Each bidirectional edge $e = \langle u, v\rangle \in E$ has weight b_e. Here, a_i are non-negative integers, and b_e are integers.

You can perform at most $4n$ operations. For each operation, select two vertices X and Y, and a non-negative integer W. Consider the shortest path from X to Y (a path is shortest if the number of edges k in it is minimum possible). Let this path consist of $k+1$ vertices $(v_0, v_1, v_2, \ldots, v_k)$ where $v_0 = X$, $v_k = Y$, and for $0 \le i < k$, the edges $e_i = \langle v_i, v_{i+1}\rangle \in E$. The operation changes the weights as follows:

$$a_X \leftarrow a_X \bigoplus W; \quad a_Y \leftarrow a_Y \bigoplus W; \quad b_{e_i} \leftarrow b_{e_i} + (-1)^i \cdot W \text{ for } 0 \le i < k.$$

Here, $\bigoplus$ denotes the bitwise XOR operation. We can notice that, if $X = Y$, nothing will change.

You need to decide whether it is possible to make all a_i and all b_e equal to 0. If it is possible, find a way to do so.

Input

The first line contains an integer T ($1 \le T \le 250$), the number of test cases. Then T test cases follow.

The first line of each test case contains a single integer n ($1 \le n \le 10^4$), the number of vertices.

The second line contains n non-negative integers a_i ($0 \le a_i < 2^{30}$), the weight on each vertex.

Then $n-1$ lines follow, each of them contains three integers u_j, v_j, w_j ($1 \le u_j, v_j \le n$, $-10^9 \le w_j \le 10^9$), representing an edge between vertices u_j and v_j with weight w_j. It is guaranteed that the given edges form a tree.

It is guaranteed that $\sum n \le 10^5$.

Output

For each test case, output "YES" on the first line if you can make all a_i and all b_e equal to 0 with no more than $4n$ operations. Output "NO" otherwise.

If you can make all weights equal to 0, output your solution in the following $k+1$ ($0 \le k \le 4n$) lines as follows.

On the next line, print an integer k: the number of operations you make.

Then print k lines, each line containing three integers X, Y, and W ($1 \le X, Y \le n$, $0 \le W \le 10^{14}$), representing one operation.

If there are several possible solutions, print any one of them.

Example

standard input	standard output
3 1 0 2 2 3 1 2 -2 3 5 4 1 1 2 -5 2 3 -5	YES 0 NO YES 3 1 3 5 2 3 7 2 3 3

Problem E. Median

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Mr. Docriz has n different kinds of objects indexed by $1, 2, \dots, n$. An object of the i-th kind weighs i kilograms. For each i, Mr. Docriz has b_i objects of the i-th kind. Among those objects, there are a_i precious objects, and the remaining ones are common objects. Now, he wants to divide all his objects into some (one or more) disjoint sets. These sets have to satisfy the following conditions:

1. Each object should go to exactly one set.
2. Each set should contain exactly one precious object.
3. In each set, the weight of the precious object should be the median weight of this set.

Please tell him whether it is possible.

For a set of size k, if we sort its elements by non-descending weight as $c_1, c_2, \dots, c_k$, the median weight of this set is defined as the weight of $c_{\lfloor (k+1)/2 \rfloor}$.

Input

The first line contains an integer T ($1 \le T \le 1000$), the number of test cases. Then T test cases follow.

The first line of each test case contains one integer n ($1 \le n \le 10^6$), specifying how many different kinds of objects Mr. Docriz has.

Then n lines follow. The i-th of these lines contains two integers a_i and b_i ($0 \le a_i \le b_i \le 10^9$), indicating that there are a_i precious objects of the i-th kind, and b_i objects of the i-th kind in total.

It is guaranteed that $\sum n \le 2 \cdot 10^6$.

Output

For each test case, output "`YES`" if it is possible to achieve the goal, or "`NO`" otherwise.

Example

standard input	standard output
3 4 1 1 1 1 1 1 1 1 4 1 1 0 1 1 1 1 1 4 0 1 1 1 1 1 1 1	YES YES NO

Problem G. Power Station of Art

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

As a modern art lover, Nocriz loves going to the Power Station of Art.

Currently, art pieces of a modern genre of art called "red-black graphs" are exhibited in the Power Station of Art. Every red-black graph art piece is an undirected labeled graph with numbers and colors associated with each vertex. Each vertex is either red or black.

It is possible to modify a graph in the following way: choose an edge and swap the numbers written on the corresponding vertices. In addition, if the colors of the two vertices are the same, the colors of both vertices are changed (from red to black or from black to red). Otherwise, the colors of the two vertices remain unchanged.

Now, Nocriz is studying two art pieces. The graphs are the same but the numbers and colors may be different. Is it possible to make some (possibly zero) modifications to the art pieces to make them be the same?

Input

The first line contains an integer T ($1 \le T \le 3 \cdot 10^4$), the number of test cases. Then T test cases follow.

The first line of each test case contains two integers n and m ($1 \le n \le 10^6$, $0 \le m \le 10^6$), the number of vertices and edges.

Then m lines follow, each of them contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$) representing an edge. It is guaranteed that there are no multiple edges in the input, and the graph **may** be unconnected.

Then numbers and colors of the two graphs follow. For each graph:

The first line contains n integers, the i-th integer a_i ($0 \le a_i \le 10^6$) representing the number written on the i-th vertex of the graph.

The second line contains n characters. If the i-th character is 'R', the i-th vertex is red. If the i-th character is 'B', the i-th vertex is black.

It is guaranteed that $\sum n \le 10^6$ and $\sum m \le 10^6$.

Output

For each test case, output a single line containing "YES" if it is possible to make some (possibly zero) modifications to the art pieces to make them be the same, or "NO" otherwise.

Example

standard input	standard output
3 2 1 1 2 3 4 RR 4 3 BB 3 2 1 2 2 3 1 1 1 RBR 1 1 1 BBB 3 3 1 2 2 3 3 1 1 1 1 RBR 1 1 1 BBB	YES NO YES

Problem K. Array

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Koishi gives you an integer array B of length n satisfying $1 \le B_1 \le B_2 \le \ldots \le B_n \le n+1$.

Let $S(T)$ denote the set of numbers that appear in array T. Koishi asks you whether an array A of length n exists such that, for any l and r such that $1 \le l \le r \le n$, the equality $S(A[l,r]) = S(A[1,n])$ holds if and only if $r \ge B_l$. If so, please construct an array A that satisfies the condition above.

Here, $A[l,r]$ represents the sub-array of A formed by $A_l, A_{l+1}, \ldots, A_r$.

You can only use integers from 0 to 10^9 in the array. It can be shown that, if a solution exists, then there also exists a solution satisfying this condition.

Notice: If there exists such an index i ($1 \le i \le n$) that $B_i < i$ holds, the required A must not exist.

Input

The first line contains an integer T ($1 \le T \le 6 \cdot 10^4$), the number of test cases. Then T test cases follow.

The first line of each test case contains an integer n ($1 \le n \le 2 \cdot 10^5$), the length of array B (and A).

The next line contains n integers $B_1, B_2, \ldots, B_n$ ($1 \le B_1 \le B_2 \le \ldots \le B_n \le n+1$), the array that Koishi gives you.

It is guaranteed that $\sum n \le 2.6 \cdot 10^6$.

Output

For each test case, print one line. If such an array A doesn't exist, output -1. Otherwise, you should output n numbers: the array A consisting of integers in the range from 0 to 10^9. If there are several possible solutions, print any one of them.

Example

standard input	standard output
3 4 3 3 5 5 7 4 6 6 7 8 8 8 5 2 3 4 4 6	2 2 1 1 2 3 4 1 3 2 4 -1

Problem L. A Post Robot

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 seconds
Memory limit:	512 mebibytes

DT is a big fan of digital products. He writes posts about technological products almost everyday in his blog.

But there is such few comments of his posts that he feels depressed all the day. As his best friend and an excellent programmer, DT asked you to help make his blog look more popular. He is so warm that you have no idea how to refuse. But you are unwilling to read all of his boring posts word by word. So you decided to write a script to comment below his posts automatically.

After observation, you found words "Apple" appear everywhere in his posts. After your counting, you concluded that "Apple", "iPhone", "iPod", "iPad" are the most high-frequency words in his blog. Once one of these words were read by your smart script, it will make a comment "MAI MAI MAI!", and go on reading the post.

In order to make it more funny, you, as a fan of Sony, also want to make some comments about Sony. So you want to add a new rule to the script: make a comment "SONY DAFA IS GOOD!" when "Sony" appears.

Input

A blog article described above, which contains only printable characters(whose ASCII code is between 32 and 127), CR(ASCII code 13, '`\r`' in C/C++), LF(ASCII code 10, '`\n`' in C/C++), please process input until EOF. Note all characters are **case sensitive**.

The size of the article does not exceed 8KB.

Output

Output should contains comments generated by your script, one per line.

Example

standard input	standard output

`Apple bananaiPad lemon ApplepiSony 233`
`Tim cook is doubi from Apple iPhoneipad`
`iPhone30 is so biiiiiiig`
`Microsoft makes good App.`
	`MAI MAI MAI!`
	`MAI MAI MAI!`
	`MAI MAI MAI!`
	`SONY DAFA IS GOOD!`
	`MAI MAI MAI!`
	`MAI MAI MAI!`
	`MAI MAI MAI!`

Problem M. Colors and Pearls

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Lee has a string of n pearls. In the beginning, all the pearls have no color. He plans to color the pearls to make it more fascinating. He drew his ideal pattern of the string on a paper and asks for your help.

In each operation, he selects some continuous pearls and all these pearls will be painted to **their target colors**. When he paints a string which has k different target colors, Lee will cost k^2 points.

Now, Lee wants to cost as few as possible to get his ideal string. You should tell him the minimal cost.

Input

There are multiple test cases. Please process till EOF.

For each test case, the first line contains an integer n ($1 \le n \le 5 \times 10^4$), indicating the number of pearls. The second line contains $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) indicating the target color of each pearl.

Output

For each test case, output the minimal cost in a line.

Example

standard input	standard output
3 1 3 3 10 3 4 2 4 4 2 4 3 2 2	2 7

Problem N. Easy Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Here is a game for two players. The rule of the game is described below:

- In the beginning of the game, there are a lot of piles of beads.
- Players take turns to play. Each turn, player choose a pile i and remove some (at least one) beads from it. Then he could do nothing or split pile i into two piles with a beads and b beads.($a, b > 0$ and $a + b$ equals to the number of beads of pile i after removing)
- If after a player's turn, there is no beads left, the player is the winner.

Suppose that the two players are all very clever and they will use optimal game strategies. Your job is to tell whether the player who plays first can win the game.

Input

There are multiple test cases. Please process till EOF.

For each test case, the first line contains a postive integer $n(n \leq 10^5)$ means there are n piles of beads. The next line contains n postive integers, the i-th postive integer $a_i(a_i < 2^{31})$ means there are a_i beads in the i-th pile.

Output

For each test case, if the first player can win the game, ouput "Win" and if he can't, ouput "Lose".

Example

standard input	standard output
1 1 2 1 1 3 1 2 3	Win Lose Lose

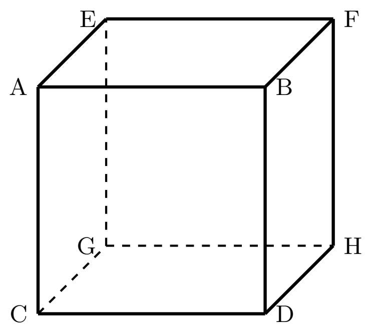
Problem O. Faces and Dices

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

There are 2 special dices on the table. On each face of the dice, a distinct number was written. Consider $a_1.a_2, a_3, a_4, a_5, a_6$ to be numbers written on top face, bottom face, left face, right face, front face and back face of dice A. Similarly, consider $b_1.b_2, b_3, b_4, b_5, b_6$ to be numbers on specific faces of dice B. It's guaranteed that all numbers written on dices are integers no smaller than 1 and no more than 6 while $a_i \neq a_j$ and $b_i \neq b_j$ for all $i \neq j$. Specially, sum of numbers on opposite faces may not be 7.

At the beginning, the two dices may face different(which means there exist some i, $a_i \neq b_i$). Ddy wants to make the two dices look the same from all directions(which means for all i, $a_i = b_i$) only by the following four rotation operations.(Please read the picture for more information)

- Left Rotation: rotate the dice by 90 degrees on $\overline{CG}$ after which $\overline{ACGE}$ becomes the bottom face.
- Right Rotation: rotate the dice by 90 degrees on $\overline{DH}$ after which $\overline{BDHF}$ becomes the bottom face.
- Front Rotation: rotate the dice by 90 degrees on $\overline{CD}$ after which $\overline{ABCD}$ becomes the bottom face.
- Back Rotation: rotate the dice by 90 degrees on $\overline{GH}$ after which $\overline{EFGH}$ becomes the bottom face.



Now Ddy wants to calculate the minimal steps that he has to take to achieve his goal.

Input

There are multiple test cases. Please process till EOF.

For each case, the first line consists of six integers $a_1, a_2, a_3, a_4, a_5, a_6$, representing the numbers on dice A.

The second line consists of six integers $b_1, b_2, b_3, b_4, b_5, b_6$, representing the numbers on dice B.

Output

For each test case, print a line with a number representing the answer. If there's no way to make two dices exactly the same, output -1.

Example

standard input	standard output
`1 2 3 4 5 6`	`0`
`1 2 3 4 5 6`	`3`
`1 2 3 4 5 6`	`-1`
`1 2 5 6 4 3`	
`1 2 3 4 5 6`	
`1 4 2 5 3 6`	

Problem P. Happiness and Cities

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

There are m visitors coming to visit country A, and they plan to visit all n cities in the country one after another. The cities are numbered from 1 to n by the order they are visited. The visitors start their tour at city 1. Each day, for each visitor i, he has p_i probability to go to next city(which means city number increases by 1), and $1 - p_i$ probability to fall in love with current city and stay there till the end of tour. If a visitor reach city n, he will not move any more.

When visitor i reach city j, he get h_{ij} units of happiness. For $j > 1$, suppose city j is visited by $c_j (c_j > 0)$ visitors and city $j-1$ is visited by $c_{j-1} (c_{j-1} > 0)$ visitors, then each of city j's visitors will get extra $\frac{c_j}{c_{j-1}} h_{ij}$ units of happiness.

Let h_{tot} denote the total happiness of all visitors at the end of tour. Now you need to calculate the expectation of h_{tot}.

Input

There are multiple test cases. Please process till EOF.

For each case, the first line contains two integers m and n $(1 \le m, n \le 16)$, indicating the number of visitors and the number of cities respectively.

The second line contains m real numbers $p_i (0 \le p_i \le 1)$—the probability for the i-th visitor to move to next city each day. The probabilities are given with at most 6 digits after decimal point.

Then there are m lines follow, each line contains n integers. The j-th integer of ith line denotes h_{ij} $(1 \le h_{ij} \le 100)$.

Output

For each test case, print a single real number in a line, represents the expectation of h_{tot}. The answer will be considered valid if it differs from the correct one by at most 10^{-5}.

Example

standard input	standard output
3 1 0.1 0.2 0.3 10 20 30 3 3 0.5 0.5 0.5 1 1 1 1 1 1 1 1 1 4 4 0.1 0.4 0.2 0.3 7 2 18 10 2 6 9 5 4 4 19 17 7 3 13 17	60.000000 6.8437500 34.23064559