Problem Peter Parker. Parallel Universes

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	2 seconds
Memory limit:	512 megabytes

Berlandia — a country with a highly developed road system. There are a total of n cities in Berlandia, and there is exactly one road between each pair of cities, accessible for travel in both directions.

For the purpose of saving electricity, only m_1 roads are illuminated in Berlandia, the i-th of which connects cities v_i and u_i. For safety reasons, it is forbidden to travel on unlit roads in Berlandia.

In a parallel universe, there is a similar country called Cherlandia, consisting of n cities. There is also exactly one road between each pair of cities in Cherlandia. The countries differ only in their electricity economy: in Cherlandia, m_2 roads are illuminated, the i-th of which connects cities a_i and b_i. It is known that in Cherlandia, it is possible to travel from any city to any other using only illuminated roads.

You possess a secret spell that allows you to choose any two different cities x and y and change the illumination on the road between cities x and y in both universes. That is, in each universe, if the road was not illuminated, it becomes illuminated, and vice versa.

You want to use this spell no more than n times in order to make it possible to travel from any city to any other in Berlandia using only illuminated roads. At the same time, after each spell is cast, Cherlandia must remain *connected*, that is, there should not exist two cities between which it is impossible to travel on illuminated roads.

Determine if this can be achieved, and if so, find a suitable sequence of spells.

Input

Each test consists of several sets of input data. The first line contains two integers t and g ($1 \le t \le 60\,000$, $0 \le g \le 10$) — the number of sets of input data and the test group number. This is followed by descriptions of the sets of input data.

The first line of each set of input data description contains three integers n, m_1, and m_2 ($3 \le n \le 300\,000$, $0 \le m_1, m_2 \le 300\,000$, $m_1, m_2 \le \frac{n(n-1)}{2}$) — the number of cities, the number of illuminated roads in Berlandia, and the number of illuminated roads in Cherlandia.

The next m_1 lines contain descriptions of illuminated roads in Berlandia. The i-th line contains two integers v_i and u_i ($1 \le v_i, u_i \le n$) — the numbers of cities connected by the i-th illuminated road. It is guaranteed that all roads are distinct.

The next m_2 lines contain descriptions of illuminated roads in Cherlandia. The i-th line contains two integers a_i and b_i ($1 \le a_i, b_i \le n$) — the numbers of cities connected by the i-th illuminated road. It is guaranteed that all roads are distinct, and that in Cherlandia, there exists a path consisting only of illuminated roads between any two cities.

Let N, M_1, and M_2 be the sum of n, m_1, and m_2 for all sets of input data in one test. It is guaranteed that $N, M_1, M_2 \le 300\,000$.

Output

For each set of input data, output "`No`" (without quotes) if there is no sequence of spells that satisfies all conditions.

Otherwise, output "`Yes`". On the second line, output an integer k ($0 \le k \le n$) — the number of spells you have used.

Then output k lines. In the i-th line, output two integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$) — the numbers of cities to which the i-th spell is applied. Note that after each spell is cast, Cherlandia must remain *connected*.

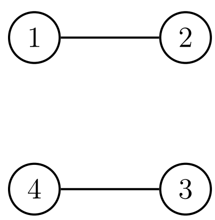
Example

input	output
3 0 3 0 3 1 2 2 3 1 3 4 2 3 1 2 3 4 1 3 1 4 2 3 4 3 3 1 2 2 3 1 3 1 4 2 4 3 4	No Yes 1 2 4 Yes 2 1 2 4 2

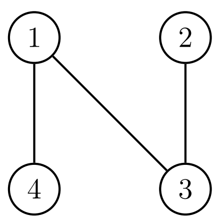
Note

In the first set of input data, there is no suitable sequence of spells, so the answer is "`No`".

In the second set of input data, the illuminated roads initially have the following structure:

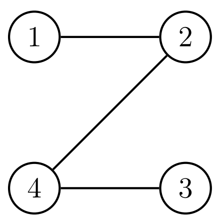


Berlandia

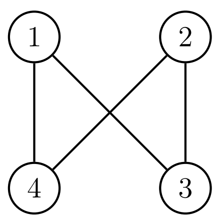


Cherlandia

After casting a spell on cities 2 and 4 in both Berlandia and Cherlandia, this road becomes illuminated, as it was unlit in both countries. After this operation, the countries will have the following structure:



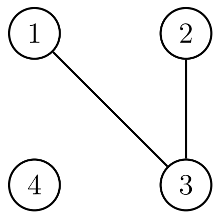
Berlandia



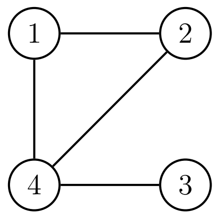
Cherlandia

After this operation, it is possible to travel from any city to any other in Berlandia, so this sequence of spells is correct.

In the third set of input data, after casting a spell on cities 1 and 2, the road between these two cities in Berlandia ceases to be illuminated, as it was illuminated before. In Cherlandia, on the contrary, the road becomes illuminated. After this operation, the countries will have the following structure:

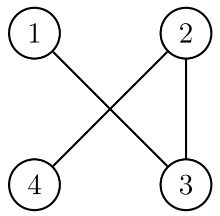


Berlandia

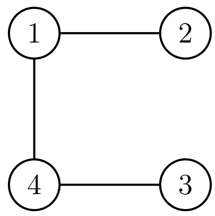


Cherlandia

After casting a spell on cities 2 and 4, the countries will have the following structure:



Berlandia



Cherlandia

Scoring

The tests for this problem consist of ten groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

Group	Points	Additional constraints	Required Groups	Comment
		N, M_1, M_2		
0	0	–	–	Examples.
1	9	$N, M_1, M_2 \le 3000$	–	$n \le 5$
2	7	$N, M_1, M_2 \le 3000$	–	$m_2 = \frac{n(n-1)}{2}$
3	10	$N, M_1, M_2 \le 3000$	–	Berlandia consists of two *connected components*[1].
4	11	$N, M_1, M_2 \le 3000$	–	There are no *isolated*[2] cities in Berlandia.
5	15	$N, M_1, M_2 \le 3000$	–	$m_2 = n - 1$ $a_i = 1$ and $b_i = i + 1$ for all $1 \le i \le n - 1$
6	8	$N, M_1, M_2 \le 3000$	5	$m_2 = n - 1$
7	12	$N, M_1, M_2 \le 3000$	–	In both countries, the road between cities 1 and 2 is illuminated.
8	6	$N, M_1, M_2 \le 3000$	0 – 7	
9	8	–	–	$m_2 = n - 1$ $a_i = i$ and $b_i = i + 1$ for all $1 \le i \le n - 1$
10	14	–	0 – 9	**Offline-evaluation.**

[1] *Connected component* — a set of cities such that there exists a path consisting only of illuminated roads between any pair of them.

[2] A city is called *isolated* if there is no illuminated road connecting it to any other city.

Problem Vito Corleone. Three Arrays

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	512 megabytes

You are given three arrays D, L, and R of length n, with elements indexed from 1, as well as the integers a_0 and b_0. You construct two arrays A and B of length $n+1$ according to the following rules:

1. $A_0 = a_0$, $B_0 = b_0$
2. For all i from 1 to n, perform the following actions:
   (a) Set the elements as $A_i = A_{i-1} + D_i$ and $B_i = B_{i-1} + D_i$.
   (b) Choose exactly **one** of the following operations and apply it:
   - $A_i = \min(A_i, L_i)$
   - $B_i = \min(B_i, R_i)$

You want to construct arrays A and B to maximize the value of $A_n + B_n$. Find the maximum value of $A_n + B_n$ that can be obtained by performing the described actions.

Input

The first line contains a single integer n ($1 \le n \le 100\,000$) — the length of arrays D, L, and R.

The second line contains n integers $D_1, D_2, \ldots, D_n$ ($0 \le D_i \le 10^9$) — the array D.

The third line contains n integers $L_1, L_2, \ldots, L_n$ ($0 \le L_i \le 10^9$) — the array L.

The fourth line contains n integers $R_1, R_2, \ldots, R_n$ ($0 \le R_i \le 10^9$) — the array R.

The fifth line contains two integers a_0 and b_0 ($0 \le a_0, b_0 \le 10^9$).

Output

Output a single integer — the maximum possible value of $A_n + B_n$ among all possible ways to construct arrays A and B.

Scoring

The tests for this problem consist of six groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

Group	Points	Additional constraints		Required groups	Comment
		n	D_i		
0	0	–	–	–	Examples.
1	13	$n \le 15$	–	0	
2	18	$n \le 300$	–	0, 1	
3	14	$n \le 5000$	$D_i = 0$	–	
4	16	$n \le 5000$	–	0–3	
5	19	–	$D_i = 0$	3	
6	20	–	–	0–5	**Offline-evaluation.**

Example

input	output
5 4 0 7 0 8 10 5 3 7 7 8 5 9 2 23 4 8	34

Note

In the first set of input data, the following sequence of actions leads to the maximum answer:

1. $A_0 = 4$, $B_0 = 8$.
2. $A_1 = A_0 + D_1 = 4 + 4 = 8$, $B_1 = B_0 + D_1 = 8 + 4 = 12$.
3. The minimum is applied to $A_1 = \min(A_1, L_1) = \min(10, 8) = 8$, the value of $B_1 = 12$ remains the same.
4. $A_2 = A_1 + D_2 = 8 + 0 = 8$, $B_2 = B_1 + D_2 = 12 + 0 = 12$.
5. The minimum is applied to $A_2 = \min(A_2, L_2) = \min(5, 8) = 5$, the value of $B_2 = 12$ remains the same.
6. $A_3 = A_2 + D_3 = 12$, $B_3 = B_2 + D_3 = 19$.
7. The minimum is applied to $A_3 = \min(A_3, L_3) = 3$, the value of $B_3 = 19$ remains the same.
8. $A_4 = A_4 + D_3 = 3$, $B_4 = B_3 + D_4 = 19$.
9. The minimum is applied to $A_4 = \min(A_4, L_4) = 3$, the value of $B_4 = 19$ remains the same.
10. $A_5 = A_5 + D_4 = 11$, $B_5 = B_4 + D_5 = 27$.
11. The value of $A_5 = 11$ remains the same, $B_5 = \min(B_5, R_5) = \min(27, 23) = 23$.
12. $A_5 + B_5 = 11 + 23 = 34$.

It can be shown that this is the maximum value.

Problem Tyler Durden. Burenka and Pether

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	3 seconds
Memory limit:	1024 megabytes

Once upon a time, the princess of Burlyandia, Burenka, decided to please her friend ReLu. Knowing that ReLu shares her interest in cryptocurrency, Burenka decided to create her own blockchain cryptocurrency called `Pether`.

After taking courses and training from an expert coach in personal growth in cybersecurity, Burenka decided that the currency `Pether` should be protected in the best possible way. As a result, due to incredibly complex and convoluted restrictions, not all users can exchange `Pether` with each other.

The structure of the `Pether` blockchain currency is indeed complex and convoluted. All users are numbered with integers from 1 to n. Each user is assigned a **unique** identifier a_i. Also, the currency has a security parameter d.

User i can directly transfer currency to user j only if $i < j$ and $a_i < a_j$. But that's not enough! Direct currency transfer between users occurs through a chain of transactions involving some number of intermediate users. During each transaction, the number of each subsequent intermediate user (including the last user j) must increase, but not by more than d. Also, all intermediate users except i and j must have an identifier strictly less than a_i.

More formally, user i can directly transfer cryptocurrency to user j if the following conditions are met:

1. It is satisfied that $i < j$
2. It is satisfied that $a_i < a_j$
3. There exists a sequence of intermediate users x of length k such that:
   (a) $i = x_1 < x_2 < \ldots < x_{k-1} < x_k = j$
   (b) For all $1 \leq t \leq k-1$, it is true that $x_{t+1} - x_t \leq d$
   (c) For all $2 \leq t \leq k-1$, it is true that $a_{x_t} < a_i$

Burenka asks you, her acquaintance programmer, to understand this system and find out for some pairs of users how to transfer `Pether` to each other.

You need to answer q queries. In each query, you need to determine whether there is a sequence of direct currency transfers (possibly through intermediate users) that allows transferring `Pether` from user u_i to user v_i. In some queries, it is also necessary to minimize the number of direct currency transfers in the process of sending currency from u_i to v_i. Please note that it is not necessary to minimize the number of transactions during each direct transfer.

Input

The first line contains three integers n, d, and g ($1 \leq n, d \leq 300\,000, 0 \leq g \leq 12$) — the number of users, the security parameter, and the test group number.

The second line contains n integers $a_1,\ a_2,\ \ldots,\ a_n$ ($1 \leq a_i \leq n$) — user identifiers. It is guaranteed that all numbers a_i are **distinct**.

The third line contains a single integer q ($1 \leq q \leq 300\,000$) — the number of queries.

The next q lines contain three integers each $t_i,\ u_i,\ v_i$ ($t_i \in \{1, 2\}, 1 \leq u_i < v_i \leq n$), where u_i is the user who should transfer the currency, and v_i is the user who should receive the currency. If $t_i = 1$, then it is necessary to determine whether it is possible to transfer the currency, and if $t_i = 2$, then it is also necessary to minimize the number of direct currency transfers.

Output

Output q lines, where the i-th line should contain the answer to the i-th query.

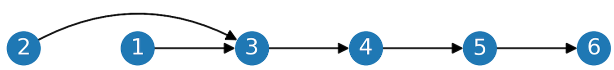
If it is not possible to transfer the currency from user u_i to user v_i, then output 0 as the answer to the i-th query. Otherwise, if $t_i = 1$, output 1, and if $t_i = 2$, output the minimum number of direct currency transfers required to transfer `Pether` from u_i to v_i.

Examples

input	output
6 1 0 2 1 3 4 5 6 6 2 1 3 2 1 2 1 1 4 2 1 5 2 1 6 1 2 6	1 0 1 3 4 1
6 2 0 1 2 3 4 5 6 6 2 1 5 2 2 5 2 1 6 2 2 6 2 1 4 2 2 4	2 2 3 2 2 1
10 2 0 2 1 4 3 5 6 8 7 10 9 10 2 1 5 1 2 5 2 3 5 2 1 9 2 5 8 2 3 9 2 1 8 1 1 2 2 3 8 2 1 9	2 1 1 4 2 3 3 0 2 4

Note

In the first example, the following direct transfers between users are possible:



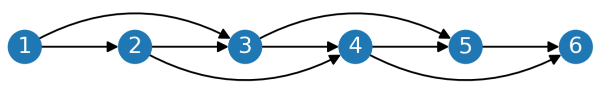
In the first query, user with index 1 can directly transfer `Pether` to user with index 3, making 2 transactions through intermediate user 2.

In the second query, a direct transfer between users with indices 1 and 2 is not possible, as $a_1 = 2 > a_2 = 1$.

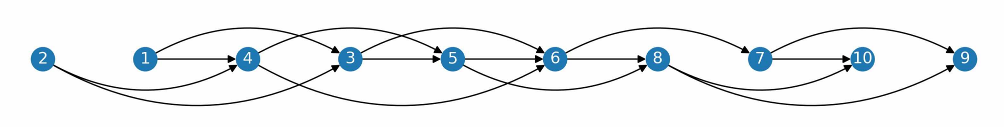
In the third query, it is possible to transfer the currency from user 1 to user 4 with two direct transfers, first transferring the currency from user 1 to user 3, and then from 3 to 4. Since $t_3 = 1$, it is only necessary to determine the possibility of transferring the currency, so the answer to the query is 1.

In the fourth query, it is possible to manage with three direct transfers: from 1 to 3, from 3 to 4, and from 4 to 5.

In the second example, the following direct transfers between users are possible:



In the third example, the following direct transfers between users are possible:



Scoring

The tests for this problem consist of twelve groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

Group	Points	Additional constraints			Required groups	Comment
		n	q	v_i, a_n, t_i		
0	0	–	–	–	–	Examples.
1	10	$n \leq 100$	$q \leq 100$	–	–	
2	7	$n \leq 1000$	–	–	1	
3	14	–	–	$a_n = n, v_i = n$	–	
4	10	–	$q = 1$	$v_i = n$	–	
5	9	–	–	$v_i = n$	3, 4	
6	7	–	–	$t_i = 2$	–	The answer does not exceed 10
7	7	–	–	$t_i = 2$	1, 6	The answer does not exceed 150
8	13	–	–	$t_i = 1$	–	
9	10	$n \leq 50\,000$	$q \leq 50\,000$	–	1	
10	4	$n \leq 100\,000$	$q \leq 100\,000$	–	1, 9	
11	4	$n \leq 200\,000$	$q \leq 200\,000$	–	1, 9, 10	
12	5	–	–	–	0 – 11	**Offline-evaluation.**

Problem Emmanuel Goldstein. Almost Certainly

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	512 megabytes

Let's say that two multisets are equal **almost certainly** if they are equal up to one element. That is, it should be possible to change at most one element in the first multiset so that they become equal. For example, the multisets $\{1, 1, 2\}$ and $\{1, 2, 3\}$ are equal *almost certainly*, $\{1, 1, 1\}$ and $\{1, 1, 1\}$ are equal *almost certainly*, and $\{1, 2, 3\}$ and $\{3, 4, 5\}$ are not equal *almost certainly*.

A boy named Vasya really liked this definition and immediately came up with a problem about it.

Vasya has two arrays a and b, where $a_i \geq b_i$ for all i from 1 to n. Vasya can apply the following operation to array a as many times as he wants (possibly zero times): choose any index i ($1 \leq i \leq n$) and subtract 1 from a_i. At the same time, Vasya does not change array b.

Vasya quickly understood what sequence of operations is needed to make the multiset of values of arrays a and b equal *almost certainly*. Therefore, Vasya made the task more complicated — now for each prefix of these arrays, he wants to know the minimum number of operations needed to make the prefixes of the arrays equal *almost certainly*.

More formally, for each k from 1 to n, Vasya wants to take the elements $a_1, a_2, \ldots, a_k$, as well as the elements $b_1, b_2, \ldots, b_k$. Vasya wants to know the minimum number of operations needed to make the multisets of these elements equal *almost certainly*. Note that the task for each k is solved **independently**.

Input

Each test consists of one or more sets of input data. The first line contains a single integer t ($1 \leq t \leq 100\,000$) — the number of sets of input data. Then follows the description of the sets of input data.

The first line of each set of input data contains a single integer n ($1 \leq n \leq 200\,000$) — the size of arrays a and b.

The second line of each set of input data contains n integers $a_1,\ a_2,\ \ldots,\ a_n$ ($1 \leq a_i \leq 10^9$) — the elements of array a.

The third line of each set of input data contains n integers $b_1,\ b_2,\ \ldots,\ b_n$ ($1 \leq b_i \leq a_i$) — the elements of array b.

Let N be the sum of n for all sets of input data in one test. It is guaranteed that $N \leq 200\,000$.

Output

For each set of input data, output n integers, each of which is the answer to the task for each possible prefix length. It can be shown that the answer always exists.

Examples

input	output
4 2 3 4 1 2 2 3 4 1 3 3 11 17 14 1 13 10 4 100 11 50 42 30 1 20 5	0 1 0 0 0 4 2 0 10 30 48
3 4 2 4 5 12 1 3 4 10 4 3 5 8 20 1 2 6 7 4 4 4 4 4 1 2 3 4	0 1 1 3 0 1 3 6 0 2 3 3

Note

Consider the first set of input data in the first example.

- For a prefix of length 1, nothing needs to be done.
- For a prefix of length 2, $a_1 = 3$ needs to be decreased by 1 once, after which a will be equal to $[2, 4]$, b will be equal to $[1, 2]$, and they will be equal *almost certainly.*

Consider the third set of input data in the first example.

- For a prefix of length 1, nothing needs to be done.
- For a prefix of length 2, $a_2 = 17$ needs to be decreased by 4 times, after which the prefix of a will be equal to $[11, 13]$, the prefix of b will be equal to $[1, 13]$, and they will be equal *almost certainly.*
- For a prefix of length 3, $a_1 = 11$ needs to be decreased by 1 and $a_3 = 14$ needs to be decreased by 1, after which a will be equal to $[10, 17, 13]$, b will be equal to $[1, 13, 11]$, and they will be equal *almost certainly.*

Scoring

The tests for this problem consist of six groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

Group	Points	Additional constraints	Required groups	Comment
		N		
0	0	–	–	Examples.
1	16	$N \leqslant 100$	0	–
2	13	$N \leqslant 500$	0, 1	–
3	24	$N \leqslant 3000$	0 – 2	–
4	13	–	–	$a_i < b_{i+1}$
5	14	–	4	$a_i \leqslant a_{i+1}, b_i \leqslant b_{i+1}$
6	20	–	0 – 5	**Offline-evaluation.**

It can be shown that all tests of the fourth group satisfy the constraints of the fifth group.