

Problem A. Standard Othman

Input file: standard input
Output file: standard output
Balloon Color: Black

Othman is a lazy student who relies entirely on leaked exam statistics to pass his high school national exams with a high score.

This morning, Othman is taking his final Biology exam, which consists of 100 multiple-choice questions. Each question has four possible choices: a, b, c, and d.

By pure luck (or so he claims), Othman found out that among the 100 questions, exactly A questions have the correct answer a, B questions have the correct answer b, C questions have the correct answer c, and D questions have the correct answer d. However, he has no idea which specific answer belongs to which question.

To make use of this information, Othman decides to mark exactly A questions as a, B questions as b, C questions as c, and D questions as d on his answer sheet.

Help Othman determine the maximum number of questions he is **guaranteed** to answer correctly, regardless of how the actual correct answers are arranged.

Input

The first line contains a single integer t ($1 \leq t \leq 1000$) — the number of test cases.

Each test case consists of a single line containing four non-negative integers A , B , C , and D ($0 \leq A, B, C, D \leq 100$, $A + B + C + D = 100$) — the counts of options a, b, c, and d, respectively.

Output

For each test case, output a single integer — the maximum number of correct answers Othman is guaranteed to get in the worst-case scenario.

Example

standard input	standard output
3	0
25 25 25 25	0
50 0 50 0	20
60 20 20 0	

Problem B. Nostalgia

Input file: standard input
Output file: standard output
Balloon Color: Light Green

Omar left his home for 14 years because of the war in Syria.

He has a favorite array a of size n .

Before he left, he used to do the following process:

- Consider all possible triples of indices (i, j, k) with $1 \leq i < j < k \leq n$.
- For each triple, he takes the three values a_i, a_j, a_k , sorts them, and writes down the **median** (the middle element) on a piece of paper.
- He does this for **every** triple, in some random order.

After returning home, he found that his array was lost due to the destruction. However, he managed to find the paper where he had written all the medians.

But the paper is damaged — some numbers may be incorrect. Omar is not sure if the information on the paper is valid or not.

Given the size n and a multiset of $\binom{n}{3}$ numbers (the numbers written on the paper), your task is to determine whether there exists **any** array a of size n such that the multiset of medians of all triples of indices matches exactly the given multiset.

Here $\binom{n}{3}$ denotes the binomial coefficient, which is the number of triples of distinct indices from n indices:

$$\binom{n}{3} = \frac{n(n-1)(n-2)}{6}$$

For example, $\binom{4}{3} = 4$, $\binom{5}{3} = 10$, and $\binom{6}{3} = 20$.

Input

The first line contains a single integer n ($3 \leq n \leq 100$).

The second line contains $\binom{n}{3}$ integers $b_1, b_2, \dots, b_{\binom{n}{3}}$ ($1 \leq b_i \leq 10^9$) — the numbers written on the paper.

Output

print "YES" if there exists an array a of length n that produces exactly this multiset of medians, otherwise print "NO".

Examples

standard input	standard output
4 7 3 3 7	YES
5 1 2 3 4 5 6 7 8 9 10	NO

Problem C. Dont Overkill

Input file: standard input
Output file: standard output
Balloon Color: Purple

Tomeh always boasts "Strings are my playground"

yet he never manages to solve the problems without resorting to overkill.

To prove that simplicity is the true mark of mastery: Given a string s , write a program that outputs the frequency count of the most frequent substring[†] within s , without using tomeh's overkill power.

Input

The first line contains a single integer T ($1 \leq T \leq 1000$), the number of test cases.

Each test case consists of two lines: The first line of each test case contains a single integer N ($1 \leq N \leq 10^5$), representing the length of the string. The second line of each test case contains a string s of length N , consisting of lowercase English letters.

It is guaranteed that the sum of N over all test cases does not exceed 10^5 .

Output

Output a single integer representing the frequency count of the most frequent substring within s .

Example

standard input	standard output
3	2
4	2
abab	2
6	
cabbca	
2	
dd	

Note

[†]A string a is a substring of a string b if a can be obtained from b by deletion of several (possibly, zero or all) characters from the beginning and several (possibly, zero or all) characters from the end.

Problem D. Where were you 14 years ago?

Input file: standard input
Output file: standard output
Balloon Color: Baby Pink

This is an interactive problem.

After the war in T-Land ended, the people were divided into three groups:

- The fighters who liberated the country, called **Thowwar**, denoted by the letter T.
- The supporters of the previous regime despite all of its crimes, called **Floul**, denoted by the letter F.
- People with no opinion or clear identity, called **Mkaw3**, denoted by the letter M.

There are n people in the waiting room. Your task is to determine the type of every person.

Fortunately, you can ask Abu Dujana for help. He is able to identify people's apparent identities by simply looking at them.

You may perform the following operation any number of times:

- Choose exactly k ($2 \leq k \leq n$) people and send them into Abu Dujana's inspection room.
- Their behavior is as follows:
 - Every **Thowwar** remains a **Thowwar**.
 - Every **Floul** remains a **Floul**.
 - If a **Mkaw3** does not find any **Thowwar** inside the room, they feel safe and return to their original nature as a **Floul**.
 - Otherwise, if there is at least one **Thowwar** in the room, every **Mkaw3** immediately starts singing "Labat Labat" and successfully fools Abu Dujana into thinking they are a **Thowwar**.

Formally, let the chosen group contain T **Thowwar**, F **Floul**, and M **Mkaw3**. Abu Dujana returns a single integer:

- If $T = 0$, Abu Dujana returns 0.
- Otherwise, Abu Dujana returns $T + M$.

Determine the type of every person while using at most **812** operations (in honor of **8/12**, T-Land Liberation Day).

It is guaranteed that among the n people there is at least one **Thowwar**, at least one **Floul** and at least one **Mkaw3**.

Input

The first line contains a single integer n ($3 \leq n \leq 400$) — the number of people.

Interaction Protocol

The first line of the input contains a single integer n ($3 \leq n \leq 400$) — the number of people.

To ask a query, print a line in the following format:

? k i1 i2 ... ik

where $2 \leq k \leq n$, all indices are distinct, and $1 \leq i_j \leq n$.

After printing a query, flush the output and read a single integer x .

Suppose the chosen group contains:

- T Thowwar,
- F Floul,
- M Mkaw3een.

The interactor responds as follows:

- If $T = 0$, it returns 0.
- Otherwise, it returns $T + M$.

You may ask at most **812** queries.

When you have determined the type of every person, print

! s

where s is a string of length n , and the i -th character is:

- T if the i -th person is a Thowwar;
- F if the i -th person is a Floul;
- M if the i -th person is a Mkaw3.

After printing the answer, terminate your program immediately.

Your program will receive the verdict **Wrong Answer** if it asks more than 812 queries or prints an invalid query.

Remember to flush the output after every query. For example, use:

- `fflush(stdout)` in C/C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- `flush(output)` in Pascal.

Example

standard input	standard output
5	? 5 1 2 3 4 5
FMTMF	? 3 1 2 5
3	? 2 1 5
0	? 2 3 5
0	! FMTMF
1	

Note

In the sample, there are $n = 5$ people. The hidden identities are FMTMF (Persons 1 and 5 are Floul, Persons 2 and 4 are Mkaw3, and Person 3 is Thowwar).

The interaction proceeds as follows:

- **Query 1** (? 5 1 2 3 4 5): You send everyone to the room. The group contains 1 Thowwar (Person 3), 2 Floul (Persons 1 and 5), and 2 Mkaw3 (Persons 2 and 4). Because there is at least one Thowwar present ($T = 1$), the Mkaw3 people act like Thowwar. Abu Dujana returns $T + M = 1 + 2 = 3$.
- **Query 2** (? 3 1 2 5): You send persons 1, 2, and 5 to the room. The group contains 2 Floul and 1 Mkaw3. Since there are absolutely no Thowwar present ($T = 0$), the Mkaw3 person feels safe and acts like a Floul. Abu Dujana returns 0.
- **Query 3** (? 2 1 5): You send persons 1 and 5. Both are Floul ($T = 0$). Abu Dujana returns 0.
- **Query 4** (? 2 3 5): You send persons 3 and 5. The group contains 1 Thowwar and 1 Floul. There are no Mkaw3 ($M = 0$). Abu Dujana returns $T + M = 1 + 0 = 1$.
- **Final Answer** (! FMTMF): Having deduced the correct types, you print the final string.

Problem E. Mesh mante2 ya zalame

Input file: standard input
Output file: standard output
Balloon Color: White

Khaled and Saleem spent months building a complex logistics network, but everything crashed right on launch day. Watching the servers collapse on the screen, Saleem slammed his hand on the desk and shouted: "Mesh mante2 ya zalame!"

To salvage what they could, Khaled realized they had to track and identify the good vertices in their directed network to correct the system's routing and avoid bankruptcy.

Their network is represented as a directed graph containing N vertices and M edges and doesn't have self loops or multiple edges, and they have a constant factor K that controls the routing cycles.

Due to continuous failures, the system faces Q queries to test the network. In each query, you are given a subset S of vertices of size c , along with two integers a and b .

For each query and for each vertex v in the graph, we define two values:

- cnt_1 : is the number of vertices u in S such that there is at least one directed **walk**[†] from u to v of length L_1 satisfying $L_1 \equiv a \pmod{K}$.
- cnt_2 : is the number of vertices u in S such that there is at least one directed **walk** from u to v of length L_2 satisfying $L_2 \equiv b \pmod{K}$.

A vertex v is considered a good vertex if cnt_1 is an odd number and cnt_2 is a **positive** even number.

For each query, help Khaled and Saleem find the number of good vertices to bring the system back online and prove that things can return to making logical sense.

Input

The first line contains three integers N , M , and K ($1 \leq N \leq 6000$, $1 \leq M \leq 50000$, $1 \leq K \leq 20$).

Each of the following M lines contains two integers u and v ($1 \leq u, v \leq N$), representing a directed edge from vertex u to vertex v .

The next line contains a single integer Q ($1 \leq Q \leq 100000$) representing the number of queries.

Each of the following Q lines describes a query. The line begins with three integers c , a , and b ($1 \leq c \leq N$, $0 \leq a, b < K$), followed by c distinct integers $s_1, s_2, \dots, s_c$ ($1 \leq s_i \leq N$) representing the vertices in the subset S .

It is guaranteed that the sum of c over all queries does not exceed 200000.

Output

For each query, print a single line containing the number of good vertices.

Example

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

Note

[†] **walk** is a sequence of vertices and edges where both vertices and edges can be visited multiple times. The length of a walk is defined as the total number of edges traversed in this sequence (counting repetitions), **not** the number of distinct edges.

Problem F. Permutation Subarrays

Input file: standard input
Output file: standard output
Balloon Color: Red

You are given an array A of size N .

You have to answer Q queries. Each query consists of two integers l and r .

For each query, count the number of contiguous subarrays that satisfy both of the following conditions:

- The length of the subarray is exactly $r - l + 1$.
- Every integer in the range $[l, r]$ appears exactly once in the subarray.

In other words, the elements of the subarray form a permutation of the integers $l, l + 1, \dots, r$.

Input

The first line contains two integers N and Q ($1 \leq N, Q \leq 2 \cdot 10^5$).

The second line contains N integers $A_1, A_2, \dots, A_N$ ($1 \leq A_i \leq 2 \cdot 10^5$).

Each of the next Q lines contains two integers l and r ($1 \leq l \leq r \leq 2 \cdot 10^5$).

Output

For each query, print a single integer — the number of contiguous subarrays whose length is exactly $r - l + 1$ and whose elements form a permutation of all integers from l to r .

Examples

standard input	standard output
5 3 1 2 4 3 5 1 3 1 4 2 5	0 1 1
11 3 1 3 2 4 2 1 3 4 3 2 1 1 4 1 3 3 4	4 3 2

Problem G. El Babies Fel Peace

Input file: standard input
Output file: standard output
Balloon Color: Navy Blue

While most of the world spends their time in disputes about (*7as* or *3rr*), (*i64* or *11*), (*VS Code* or *Geany*), a baby lives a peaceful life, saying only one word: "AGUAGU".

Now, the troublemakers want to discover the baby's bias.

They ask him a single question: is he *7as* or *3rr*? The baby responds with the only word he knows: "AGUAGU".

Input

The input contains a single string: "*7as or 3rr ?*".

Output

Print a single line containing the exact word the baby says: "AGUAGU".

Example

standard input	standard output
<i>7as or 3rr ?</i>	AGUAGU

Problem H. The Great Hall

Input file: standard input
Output file: standard output
Balloon Color: Dark Blue

After finally replacing his old MacBook with a gaming laptop, Tomeh has become addicted to video games, just as everyone expected.

He is currently playing *The Great Hall*. The game is represented by a string t and consists of k consecutive stages.

At the end of the i -th stage ($1 \leq i \leq k$), Tomeh encounters the boss of that stage. To let him pass, the boss gives him exactly one of the following choices:

- Pay exactly i gold coins.
- Present **any palindromic substring** of string t whose length is exactly i .

As a true “Shami”, Tomeh will not spend a single coin unless he is absolutely forced to.

You are given a string s , an integer k , and q queries.

Each query consists of two integers l and r . Let $t = s[l \dots r]$ denote the corresponding substring of s .

For each query, determine the minimum total number of gold coins Tomeh must pay to clear all k stages of the game played on the string t .

Input

The first line contains an integer T ($1 \leq T \leq 1000$) — the number of test cases.

For each test case:

The first line contains two integers N and K ($1 \leq K \leq N \leq 2 \cdot 10^5$) — the length of the string s and the number of stages.

The second line contains a string s of length N , consisting of lowercase English letters.

The third line contains an integer q ($1 \leq q \leq 2 \cdot 10^5$) — the number of queries.

Each of the next q lines contains two integers l and r ($1 \leq l \leq r \leq N$), describing the substring $t = s[l \dots r]$.

It is guaranteed that:

- The sum of N over all test cases does not exceed $2 \cdot 10^5$.
- The sum of q over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print q lines. For each query, output a single integer — the minimum total number of gold coins Tomeh must pay to clear all K stages when the game is played on the string t .

Example

standard input	standard output
2	0
9 5	6
abcbaffff	14
2	26
1 9	
1 5	
12 8	
fbcaaaaacbef	
2	
3 10	
4 7	

Problem I. Mortal Kombat

Input file: standard input
Output file: standard output
Balloon Color: Yellow

Khaled and his brother are having an intense Mortal Kombat match. Determined to secure a "Flawless Victory" and leave his brother no room to breathe, Khaled decides to unleash a relentless chain of special moves.

In this game, the combo system is represented as a rooted tree consisting of n nodes, where node 1 is the root. Each node i has a specific button assigned to it, represented by a lowercase English letter c_i .

To execute the special move associated with node x , a player must input the exact sequence of buttons along the simple path from the root to node x . Let this string of button presses be S_x .

For his ultimate attack, Khaled wants to execute two special moves, u and v , in a single rapid string of inputs. A sequence of button presses is considered valid if both S_u and S_v are contiguous substrings[†] of this sequence (they can overlap).

Since Khaled wants to execute the attack as quickly as possible, he needs your help to determine the minimum number of button presses required to perform both moves. You are given q queries, each consisting of two nodes u and v . For each query, output the minimum length of a valid button sequence.

[†]A string a is a substring of a string b if a can be obtained from b by deletion of several (possibly, zero or all) characters from the beginning and several (possibly, zero or all) characters from the end.

Input

The first line contains two integers n and q ($1 \leq n, q \leq 5 * 10^5$) — the number of nodes in the combo tree and the number of queries, respectively.

The second line contains a string C of length n consisting of lowercase English letters, where the i -th character represents c_i , the button assigned to node i .

Each of the next $n - 1$ lines contains two integers x and y ($1 \leq x, y \leq n$), denoting an edge between node x and node y in the tree.

Each of the next q lines contains two integers u and v ($1 \leq u \neq v \leq n$) — the nodes representing the two special moves Khaled wants to execute.

Output

For each query, print a single integer on a new line — the minimum number of button presses required to execute both special moves u and v .

Example

standard input	standard output
5 2 abaab 1 2 2 3 1 4 4 5 3 5 2 4	4 3

Note

The string of node 3 is $S_3 = \text{"aba"}$ (path $1 \rightarrow 2 \rightarrow 3$).

The string of node 5 is $S_5 = \text{"aab"}$ (path $1 \rightarrow 4 \rightarrow 5$).

The string of node 2 is $S_2 = \text{"ab"}$ (path $1 \rightarrow 2$).

The string of node 4 is $S_4 = \text{"aa"}$ (path $1 \rightarrow 4$).

For the first query($u = 3, v = 5$) : Khaled needs the strings "aba" and "aab". The shortest sequence that contains both is "aaba"(length 4).

For the second query($u = 2, v = 4$) : Khaled needs the strings "ab" and "aa". The shortest sequence that contains both is "aab"(length 3).

Problem J. Rasoulo and the Heavy Tree

Input file: standard input
Output file: standard output
Balloon Color: Silver

One day, Hosen found a massive, incredibly heavy tree with n nodes. The tree was special: each node had a value a_i written on it, forming a permutation of integers from 0 to $n - 1$.

Without warning, Hosen threw the giant tree right into Rasoulo's arms!

"Hold the tree! Don't let it fall!" Hosen shouted as he started running away. "I am going to get us some shawarma. I will calculate its paths when I get back!"

Now, Rasoulo is stuck holding the tree. His arms are shaking, his back hurts, and he is sweating. The tree is so heavy! He desperately wants to put it down. To finish the task before his arms completely give out, Rasoulo decides to calculate the answer himself.

For any simple path between node u and node v ($u < v$), we define:

- The **length** of the path as the number of edges between u and v .
- The **MEX** of the path as the smallest non-negative integer that does *not* appear on any node along the path.

Rasoulo wants to calculate the total sum of $F(u, v)$ for all valid pairs of nodes (u, v) such that $u < v$, where:

$$F(u, v) = \text{length}(u, v) \times \text{MEX}(u, v)$$

Since the answer can be very large, print it modulo $10^9 + 7$. Can you help Rasoulo calculate the answer so he can finally drop the tree?

Input

The first line contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the number of nodes in the tree.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i < n$) — the values written on the nodes. It is guaranteed that all a_i are distinct (the array a is a permutation).

Each of the next $n - 1$ lines contains two integers u and v ($1 \leq u, v \leq n, u \neq v$), denoting an edge between node u and node v . It is guaranteed that the given edges form a valid tree.

Output

Print a single integer — the sum of $F(u, v)$ for all pairs of nodes $u < v$, modulo $10^9 + 7$.

Example

standard input	standard output
4 0 2 1 3 1 2 2 3 2 4	9

Note

In the given sample, the tree has 4 nodes. The values on the nodes are: $a_1 = 0$, $a_2 = 2$, $a_3 = 1$, and $a_4 = 3$. Let's analyze all valid pairs (u, v) where $u < v$:

- **Pair (1, 2):** The path is $1 \rightarrow 2$. The values on the path are $\{0, 2\}$. The length is 1. The MEX is 1 (since 0 is present, but 1 is missing). $F(1, 2) = 1 \times 1 = 1$.
- **Pair (1, 3):** The path is $1 \rightarrow 2 \rightarrow 3$. The values on the path are $\{0, 2, 1\}$. The length is 2. The MEX is 3 (since 0, 1, 2 are all present). $F(1, 3) = 2 \times 3 = 6$.
- **Pair (1, 4):** The path is $1 \rightarrow 2 \rightarrow 4$. The values on the path are $\{0, 2, 3\}$. The length is 2. The MEX is 1. $F(1, 4) = 2 \times 1 = 2$.
- **Pair (2, 3):** The path is $2 \rightarrow 3$. The values on the path are $\{2, 1\}$. The length is 1. The MEX is 0. $F(2, 3) = 1 \times 0 = 0$.
- **Pair (2, 4):** The path is $2 \rightarrow 4$. The values on the path are $\{2, 3\}$. The length is 1. The MEX is 0. $F(2, 4) = 1 \times 0 = 0$.
- **Pair (3, 4):** The path is $3 \rightarrow 2 \rightarrow 4$. The values on the path are $\{1, 2, 3\}$. The length is 2. The MEX is 0 (since 0 is missing). $F(3, 4) = 2 \times 0 = 0$.

The total sum is $1 + 6 + 2 + 0 + 0 + 0 = 9$.

Problem K. Homsî was sleeping

Input file: standard input
Output file: standard output
Balloon Color: Brown

Homsî had a strange dream involving two positive integers k and b .

In his dream, he saw a strictly increasing infinite array a such that for every non-negative integer x , there exist exactly one ordered tuple of indices $i_1, i_2, \dots, i_k$ (where the indices are not necessarily distinct and are not required to be in non-decreasing order) satisfying

$$x = a_{i_1} + b \cdot a_{i_2} + b^2 \cdot a_{i_3} + \dots + b^{k-1} \cdot a_{i_k}$$

In other words, every non-negative integer has a unique representation of the above form.

After waking up, Homsî told his friends about the dream. Hussain doubted that such an array could even exist, so he challenged Homsî with the following task.

For each test case, given the integers b , k , l , and r , determine the value of $a_l + a_{l+1} + \dots + a_r$ modulo $10^9 + 7$.

If no array satisfying the required property exists, output -1.

Input

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

Each of the next t lines contains three integers b , k , l , and r ($2 \leq b, k \leq 10^{18}, 1 \leq l \leq r \leq 10^{18}$).

Output

For each test case, print the sum $a_l + a_{l+1} + \dots + a_r \bmod (10^9 + 7)$ if such an array exists, -1 otherwise.

Example

standard input	standard output
3	5
2 2 1 3	0
2 3 1 1	165
3 4 5 6	

Problem L. Chasing Hamoosh

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Pink`

Due to severe destruction in Syria, many residents have fled and left their homes empty. The thief Hamoosh takes advantage of this to steal iron from the abandoned buildings, but the policeman Homsy spots him and initiates a pursuit.

There are N buildings arranged in a circle, numbered 1 to N in clockwise order. For $1 \leq i < N$, the building following building i is $i + 1$, and the building following building N is 1. The i -th building contains W_i units of iron.

You must choose the initial starting building for Hamoosh and the initial starting building for Homsy. At time 0, both are placed on their respective starting buildings. In each turn, they move simultaneously in the clockwise direction: Hamoosh jumps forward by exactly A buildings, and Homsy jumps forward by exactly B buildings.

Hamoosh steals all W_i units of iron from any building i he lands on for the first time. If Hamoosh and Homsy land on the same building at the exact same moment, Hamoosh is immediately caught and the game ends. When caught, Hamoosh is arrested before he can steal the iron from that specific building.

You must assign the starting positions such that Homsy is guaranteed to eventually catch Hamoosh. Under this restriction, what is the maximum amount of iron Hamoosh can collect before being caught?

Input

The first line contains three integers N , A , and B ($1 \leq N \leq 5000$, $1 \leq A, B \leq N$) — the number of buildings, the number of buildings Hamoosh jumps per turn, and the number of buildings Homsy jumps per turn.

The second line contains N integers $W_1, W_2, \dots, W_N$ ($0 \leq W_i \leq 10^9$) — the amount of iron in each building.

Output

Print a single integer — the maximum amount of iron Hamoosh can collect before being caught.

Example

standard input	standard output
5 2 3 10 20 30 40 50	140

Problem M. Power of 3afweya

Input file: standard input
Output file: standard output
Balloon Color: Green

You are given a permutation $P = (P_1, P_2, \dots, P_N)$ of the integers from 1 to N .

For a subarray $P[l \dots r] = (P_l, P_{l+1}, \dots, P_r)$ (where $1 \leq l \leq r \leq N$), let us define its **Value** as follows:

$$\text{Value}(l, r) = \text{Sum}(l, r) - \text{MEX}(l, r)$$

Where:

- $\text{Sum}(l, r) = \sum_{k=l}^r P_k$
- $\text{MEX}(l, r)$ is the smallest **positive** integer $(1, 2, 3, \dots)$ that does **not** appear in the subarray $P[l \dots r]$.

We call a range **3afwey** if its value is as large as possible. El 3afweya is calling ..., and you have to respond. Determine the value of the 3afwey range.

Input

The first line contains a single integer T ($1 \leq T \leq 10^4$) — the number of test cases.

Each testcase is structured as follows:

The first line contains an integers N ($2 \leq N \leq 5 \cdot 10^5$) — the size of the permutation.

The second line contains N space-separated integers $P_1, P_2, \dots, P_N$ ($1 \leq P_i \leq N$) — the initial permutation.

It is guaranteed that the sum of N over all test cases does not exceed $5 \cdot 10^5$

Output

Print T integers, The i -th line should contain a single integer representing the maximum Value of a subarray of the i -th test case.

Example

standard input	standard output
2	13
5	10
1 3 2 4 5	
5	
3 1 4 2 5	

Problem N. Stable Factory Assignment

Input file: standard input
Output file: standard output
Balloon Color: Cumin

Khaled is the CEO of a rapidly growing industrial empire. His company recently acquired N new factories, and he has hired N workers to operate them. Both the workers' homes and the factories are located along a long, straight highway, which can be represented as the X -axis.

The i -th worker lives at coordinate x_i , and the j -th factory is located at coordinate y_j . Khaled needs to assign exactly one worker to each factory so that no two workers are assigned to the same factory.

Khaled requires a “stable” assignment. An assignment is invalid if swapping two workers' factories strictly increases the sum of their squared travel distances. Formally, for any worker i assigned to factory a and worker j assigned to factory b , the following condition must **NOT** hold:

$$(x_i - y_a)^2 + (x_j - y_b)^2 < (x_i - y_b)^2 + (x_j - y_a)^2$$

Given the locations, help Khaled find any valid assignment satisfying his strict stability condition.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains a single integer N ($1 \leq N \leq 3000$) — the number of workers and the number of factories.

The second line of each test case contains N distinct integers $x_1, x_2, \dots, x_N$ ($-10^9 \leq x_i \leq 10^9$) — the coordinates of the workers.

The third line of each test case contains N distinct integers $y_1, y_2, \dots, y_N$ ($-10^9 \leq y_i \leq 10^9$) — the coordinates of the factories.

It is guaranteed that the sum of N over all test cases does not exceed 3000.

Output

For each test case, output a single line containing N space-separated integers $p_1, p_2, \dots, p_N$, where p_i is the 1-based index of the factory assigned to the i -th worker.

If there are multiple valid assignments that satisfy Khaled's condition, you may output any of them.

Example

standard input	standard output
1 4 7 -5 -9 5 2 -8 7 -11	4 1 3 2