Insomnia 2025

A. XO-OR

1 second, 256 megabytes

Given two integers x and y, choose an integer $v \in [0, x]$ such that the size of a set S of distinct non-negative integers is maximized.

Formally, determine the largest possible size of a set

$$S = \{s_1, s_2, \dots, s_m\}$$

satisfying the following conditions:

- The bitwise XOR of all elements equals y:

$$s_1 \oplus s_2 \oplus \dots \oplus s_m = y.$$

- The bitwise OR of all elements equals v:

$$s_1 \mid s_2 \mid \dots \mid s_m = v.$$

Output the maximum possible value of m. If no valid set exists, output -1.

Input

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

Each test case contains two integers x and y $(0 \le x, y \le 10^{18})$.

Output

For each test case, print a single integer — the maximum possible size of the set S. If no such set can be formed, report -1.

input

```
2
2 2
3 0
```

output

```
2
4
```

- For the first test case, let $(x, y) = (2, 2)$ and choose $v = 2$. The resulting set is defined as:

$$S = \{0, 2\}$$

  - XOR of all elements: $0 \oplus 2 = 2$
  - OR of all elements: $0 \mid 2 = 2$

- For the second test case, let $(x, y) = (3, 0)$ and choose $v = 3$. The resulting set is defined as:

$$S = \{0, 1, 2, 3\}$$

  - XOR of all elements: $0 \oplus 1 \oplus 2 \oplus 3 = 0$
  - OR of all elements: $0 \mid 1 \mid 2 \mid 3 = 3$

B. Birthdays!!!

1 second, 256 megabytes

Ridham's birthday is coming up and he really loves primes and permutations. Knowing this Aastik decided to gift Ridham a permutation a of length n such that $a_i + a_{n-i+1}$ is prime for all i such that $1 \le i \le n$, but just when Aastik sat down to generate this permutation he realised that he did not know how to make such a permutation. Help Aastik to prepare the gift for Ridham.

A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

Input

The first line contains a single integer t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line and only line of each test case contains an integer n ($1 \le n \le 10^6$) — the size of the permutation a.

The sum of n over all testcases does not exceed $3 \cdot 10^6$.

Output

For each testcase, output n integers — the elements of the required permutation. If it's not possible, output -1.

input

```
2
2
3
```

output

```
1 2
2 1 3
```

C. Harmonic Grids

2 seconds, 256 megabytes

You are given a square grid h of size n that needs to be filled with digits 0 to 9. For h to be harmonic, it must follow certain rules.

The adjacent cells (cells that have a common edge) of h should have a difference of no more than 1.

Also, for any 2×2 square in h, the sum of elements on both of its diagonals should be equal. For example, $\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$ has $1 + 1 = 2 + 0$.

For any rectangle, 2 quantities – next and prev – are defined as follows.

1. next is the smallest integer greater than all elements inside the rectangle.
2. prev is the largest integer smaller than all elements inside the rectangle.

Note that for any rectangle, $\text{next} > \text{prev}$.

Any rectangle of dimension $l \times b$ is said to be a bad rectangle if the difference between the next and prev of this rectangle is equal to $l + b$.

For example, $\begin{bmatrix} 1 \end{bmatrix}$, $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$, $\begin{bmatrix} 1 & 2 & 4 \\ 3 & 4 & 2 \end{bmatrix}$ are all examples of bad rectangles, but $\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$ and $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ are good rectangles.

You need to count the number of ways to fill a harmonic grid h such that there are no bad rectangles with an area greater than k. Since the answer may be large, print it modulo $10^9 + 7$.

Input

The only input line contains two integers n $(1 \le n \le 500)$ and k $(1 \le k \le n^2)$.

Output

Print a single integer – the number of ways to fill a harmonic grid h of size n having no bad rectangles with an area greater than k.

input

```
2 1
```

output

```
10
```

input

```
2 3
```

output

```
46
```

D. Guess the permutation

2 seconds, 256 megabytes

This is an interactive problem.

In the mystical kingdom of Numeria, a powerful oracle has concealed a secret sequence — a hidden **permutation** p. The kingdom's future depends on deciphering this sequence, for it holds the key to unlocking the legendary Vault of Order, a chamber said to contain untold wisdom and treasures.

As the kingdom's greatest mathematician, you have been chosen for this task. However, the oracle does not reveal secrets so easily. Instead, she offers a challenge:

"For any query, you may choose three **distinct** indices i, j, k where $1 \le i, j, k \le n$, and I shall reveal the sum $p_i + p_j + p_k$."

Your task is to determine the hidden permutation p in no more than n queries.

Input

The first line of each testcase contains a single integer t $(1 \le t \le 10^4)$ — the number of testcases. Description of each testcase follows.

Interaction

The interaction begins with reading n $(4 \le n \le 10^4)$ — the length of hidden permutation p.

It is guaranteed that the sum of n over all testcases do not exceed 10^4.

The permutation p is fixed for each testcase. In other words, the interactor is not adaptive.

It is guaranteed that p is a permutation.

To make a query, output "`? i j k`" (without quotes, $1 \le i, j, k \le n, i \ne j \ne k \ne i$). Then you must read the response to the query, which is equal to $p_i + p_j + p_k$.

To give an answer, output "`!` p_1 p_2 p_3 `...` p_n" where $p_1, p_2, p_3 \cdots p_n$ represent the hidden permutation p.

After printing each query do not forget to output the end of line and flush* the output. Otherwise, you will get `Idleness limit exceeded verdict`. If, at any interaction step, you make an invalid query or make more queries than the given limit, then the program will terminate immediately and you will receive `Wrong Answer` verdict.

* To flush, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `sys.stdout.flush()` in Python;
- see the documentation for other languages.

input

```
2
5

8

10

6

11

11

6
```

output

```

? 5 1 2

? 1 4 3

! 3 1 2 5 4

? 4 5 6

? 3 1 4

? 5 2 3

! 6 3 1 4 2 5
```

In the first testcase, $p = [3, 1, 2, 5, 4]$.

In the second testcase, $p = [6, 3, 1, 4, 2, 5]$.

E. Easiest Problem

1 second, 256 megabytes

You are given a binary string s of length n. The score of a string is defined as the number of indices i such that $s_i = s_{i+1}$.

The beauty of a string s is the maximum possible score across all subsequences of s. Find the beauty of string s.

Note that a subsequence of a given string is a sequence that can be derived from the given string by deleting some or no characters without changing the order of the remaining characters.

Input

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

The first line of each testcase contains a single integer n ($2 \le n \le 10^5$) — the size of the string s.

The second line contains the string s, consisting of characters $0, 1$.

The sum of n over all test cases does not exceed 10^5.

Output

For each test case, print a single integer — the beauty of string s.

input

```
3
2
01
5
10110
9
101010101
```

output

```
0
2
4
```

In second testcase, we are given $s = ”10110”$, the score of s is 1 and it's beauty is 2, when we take the 1st, 3rd and 4th element as a subsequence.

F. Permaban

1 second, 256 megabytes

You are given an array a of size n. You can perform the following operation any number of times (including zero):

- Choose a subarray (l, r), where $1 \le l \le r \le n$, and an index j such that $l \le j \le r$. Set all elements in the subarray $a[l], a[l+1], \ldots, a[r]$ to the value $a[j]$. The cost of this operation is $r - l$.

The score of the array is defined as the sum of all its elements, plus the total cost of the operations performed.

Your task is to determine the minimum possible score that can be achieved after performing any number of operations.

Input

The first line contains a single integer t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each testcase contains a single integer n ($1 \le n \le 10^5$) — the size of the array a.

The second line contains n integers a_1,a_2,...,a_n ($1 \le a_i \le 10^9$) — the elements of array a.

The sum of n over all testcases does not exceed 10^6.

Output

Output a single integer for each test case - the minimum possible score after applying the operations any number of times (including zero).

input

```
2
3
1 1 1
5
2 3 3 2 3
```

output

```
3
13
```

G. Divine Powers

2 seconds, 256 megabytes

Given an array a of length n, you need to calculate the following sum:

$$f(a) = \frac{\prod_{i=1}^{n} \frac{a_i}{\phi(a_i)}}{gcd(a_1, \ldots, a_n)} \bmod 10^9 + 7$$

Since the authors thought the above problem was too easy, you need to calculate the summation of $f(s)$ across all **non-empty** subsequences s in the array a. Formally, you need to calculate:

$$\sum_{\emptyset \neq s \subseteq a} f(s) \bmod 10^9 + 7$$

Here $\phi(k)$ denotes the euler totient function. Note that a subsequence of a given array is a sequence that can be derived from the given array by deleting some or no elements without changing the order of the remaining elements.

Input

The first line contains a single integer n ($1 \le n \le 10^6$) — the size of the array a.

The second line contains n integers a_1,a_2,...,a_n ($1 \le a_i \le 10^6$) — the elements of array a.

Output

Print a single integer — The summation of $f(s)$ across all non-empty subsequences s in the array a modulo 1000000007.

input

```
4
1 2 3 4
```

output

```
500000042
```

H. Klein Moretti's Riddle

2 seconds, 256 megabytes

To prove your worth and earn your place in the Tarot Club, you must solve Klein's Riddle. You are given an array a of size n and an integer k. Additionally, you must answer q queries. For each query, an integer x_i is provided, and you need to determine how many ways there are to select a subsequence of a with exactly k elements such that the bitwise OR of its elements equals x_i. If no such subsequence exists, the answer should be 0. Since the result can be very large, output it modulo $10^9 + 7$.

Input

The first line contains two integer n and k ($1 \le k \le n \le 10^6$) — the size of the array a and the number of elements to select, respectively.

The second line contains n integers a_1,a_2,...,a_n ($1 \le a_i \le 10^6$) — the elements of array a.

The third line contains a single integer q ($1 \le q \le 10^6$) — the number of queries.

Next, there are q lines, where the i-th line describes the i-th query. Each query contains a single integer x_i ($1 \le x_i \le 10^6$).

Output

Print q integers — the answers to the queries modulo $10^9 + 7$.

input

```
9 4
8 8 1 2 2 4 16 16 16
8
30
28
1300
31
15
16
29
35
```

output

```
12
9
0
0
4
0
6
0
```

I. Min Xor Subarray

1 s., 256 MB

A permutation of length n is defined as a sequence of integers from 0 to $n-1$, each number appearing exactly once. We define a XUM value for any permutation of length n as the SUM of XOR of all subarrays with length ≥ 1. The problem is, for a given n, you have to find some permutation for which this XUM value is minimum among all the possible permutations of length n. But we found this problem to be easy.

Hence, the problem is divided into two tasks. In task 1, you have to print n integers $(0$ to $n-1)$, a permutation for which this XUM value is minimum among all the possible permutations. In task 2, you have to print only one integer, the minimum XUM value for any permutation among all the possible permutations.

† An array b is a subarray of an array a if b can be obtained from a by deletion of several (possibly, zero or all) elements from the beginning and several (possibly, zero or all) elements from the end. In particular, an array is a subarray of itself.

Input

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

Each test case contains 2 integers c and n $(1 \leq c \leq 2\ , 1 \leq n \leq 10^9)$.

Additional constraint on the input: The sum of n over all the test cases with $c = 1$ does not exceed $2 \cdot 10^5$.

Output

For test cases with $c = 1$, you have to solve task 1. Print a permutation of n integers, 0 to $n-1$, each appearing once for which the XUM value would be minimum.

For test cases with $c = 2$, you have to solve task 2. Print 1 integer, the minimum XUM value among all the permutations of length n. Since this value can be very large, print it modulo $10^9 + 7$.

input

```
5
2 1
1 3
2 4
1 5
2 10
```

output

```
0
2 0 1
12
4 0 2 3 1
212
```

J. Alice and Bob

1 second, 256 megabytes

Alice gives you an array a of size n. You can perform the following operation any number of times (including zero):

- Choose two distinct indexes i and j, such that $a_j \leq a_i$ and replace a_i with $a_i - a_j$.

Bob being the troublemaker he is, orders you to change $\textbf{exactly one}$ element in the array before performing the above operation. Your goal is to maximize the gcd of the whole array, $gcd(a_1, a_2, \ldots, a_n)$.

Input

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

The first line of each testcase contains a single integer n ($2 \leq n \leq 10^5$) — the size of the array a.

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

The sum of n over all testcases does not exceed 10^6.

Output

Output a single integer for each testcase — the maximum possible value of $gcd(a_1, a_2, \ldots, a_n)$.

input

```
3
2
5 5
5
4 4 4 2 4
7
3 6 12 990 9 12 15
```

output

```
5
4
3
```

K. Land Distribution

2 seconds, 256 megabytes

Consider a rooted family tree having n nodes. Each node in this tree (except the root) has a unique parent. Node 1 is the root of the tree. Initially, the root owns k units of land.

Each generation, all nodes which own positive amount of land and have at least one children, have to distribute all of the land among their children. Each child must receive an integral amount of land. Also, the minimum amount of land received by any child and the maximum amount of land received by any child (of the same parent) must not differ by more than 1.

Let a_i $(2 \le i \le n)$ be the amount of land received by the node i from their parent and $a_1 = k$. Over all possible ways of distributing the land, find the maximum possible value of

$$\sum_{i=1}^{n}(a_i \oplus i)$$

Note that a_i represents the amount of land received by node i from their parent, not the final amount of land it owns.

Here, $\oplus$ denotes the bitwise XOR operator.

Input

The first line of input contains t $(1 \le t \le 100)$ — the number of testcases. Description of each testcase follows.

The first line of each testcase contains two integers n $(1 \le n \le 10^5)$ and k $(1 \le k \le 10^9)$ — the number of nodes in the tree and the initial amount of land owned by the root.

The second line of each testcase contains $n-1$ integers $p_2, p_3, \cdots, p_n$ $(1 \le p_i \le n)$, where p_i is the parent of i^{th} node.

It is guaranteed that the sum of n over all testcases don't exceed 10^5.

Output

For each testcase, output the maximum possible of $\sum_{i=1}^{n}(a_i \oplus i)$ on a separate line.

input

```
3
1 5

3 4
1 1
4 9
1 1 2
```

output

```
4
6
23
```

For the second testcase, node 1 can divide the land equally so that node 2 and 3 each recieve 2 units of land. Note that this is the only way node 1 can divide the land.

For the third testcase, node 1 may give 5 units of land to node 2 and 4 units of land to node 3.

L. Tree Harmony

1 second, 256 megabytes

In the mystical **Land of Trees**, there exists a rooted tree with n nodes. The tree is rooted at node 1. Each node is uniquely numbered from 1 to n, and the tree structure is defined by $n-1$ edges. Additionally, each node i has a value a_i.

Two nodes u and v are said to be **compatible** if they satisfy the following conditions:

1. Let L be the **Lowest Common Ancestor** (LCA) of u and v. For u and v to be compatible, u must be the LCA, i.e., $L = u$.
2. Consider the set S, which consists of all nodes in the subtree of u excluding those in the subtree of v. We must be able to partition S into two disjoint subsets s_1 and s_2 such that:
   - $s_1 \cup s_2 = S$, i.e., the union of both subsets must be equal to S,
   - $s_1 \cap s_2 = \emptyset$, i.e., the subsets must be disjoint, meaning they share no elements.
3. It must be possible to form pairs between elements of s_1 and s_2 such that each node appears in exactly one pair. A node x can be paired with a node y if and only if $a_x \ne a_y$.

You are given q queries. In each query, you are given two nodes u and v, and you must determine whether they are **compatible**.

Input

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

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^5)$ — the values associated with each node.

The next $n-1$ lines describe the tree structure. Each of these lines contains two integers x and y $(1 \le x, y \le n)$, representing an edge between nodes x and y.

The next line contains an integer q $(1 \le q \le 10^5)$ — the number of queries.

The next q lines each contain two integers u and v $(1 \le u, v \le n)$, representing a query where we must determine if nodes u and v are compatible.

Output

For each query, output `YES` if the nodes u and v are compatible; otherwise, output `NO`. The output is case-insensitive.

input

```
5
1 1 2 1 2
1 2
2 3
3 4
4 5
4
4 5
3 5
2 5
1 4
```

output

```
NO
YES
NO
NO
```

input

```
6
1 2 3 4 5 6
1 2
2 3
3 4
4 5
5 6
4
1 3
1 2
2 5
1 6
```

output

```
YES
NO
NO
NO
```

M. Alternating sum

3 seconds, 1024 megabytes

Given an array a of n integers. We define a function $f(a)$ as follows:

- $f(a)$ is the maximum possible sum of elements in a, where no two consecutive elements in a are selected.

Your task is to compute the sum of $f(s)$ over all non-empty subsequences s of the array a. Since this number can be large output it modulo 10^9+7. Formally, you need to find:

$$\sum_{\emptyset \neq s \subseteq a} f(s) \bmod 10^9+7$$

Input

The first line contains a single integer t ($1 \le t \le 10^3$). The description of the test cases follows.

The first line of each testcase contains a single integer n ($1 \le n \le 10^3$) — the size of the array a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2 \cdot 10^5$) — the elements of array a.

The sum of n over all testcases does not exceed 10^3.

Output

Output a single integer for each testcase — the sum of $f(s)$ over all non-empty subsequences s of the array a modulo 10^9+7.

input

```
3
3
3 7 4
5
1 100 5 10 5
10
1 1 2 1 1 2 1 1 1 1
```

output

```
39
1774
3568
```

N. Maximize Minimum Mex

1 second, 256 megabytes

You are given an undirected tree with n vertices, labeled from 0 to $n-1$. The tree has $n-1$ edges and is connected and acyclic.

Your task is to select exactly k vertices ($2 \le k \le n$) such that the minimum MEX value over all paths between two chosen vertices is maximized.

For any two vertices u and v, define P_{uv} as the set of vertex labels that appear on the unique simple path between them. The function $\mathrm{MEX}(P_{uv})$ is defined as the smallest non-negative integer that does not appear in P_{uv}.

Formally, for a chosen set of vertices S with $|S|=k$, we define:

$$f(S) = \min_{u,v \in S, u \neq v} \mathrm{MEX}(P_{uv}).$$

Your goal is to maximize $f(S)$ over all valid choices of S.

For each integer k from 2 to n, compute the maximum possible value of $f(S)$.

Input

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

The first line of each test case contains an integer n ($3 \le n \le 10^5$) — the number of nodes in the tree.

The next $n-1$ lines each contain two integers u and v ($0 \le u, v < n$), representing an undirected edge between nodes u and v.

It is guaranteed that the sum of all n over all test cases does not exceed 10^6.

Output

For each test case, print $n-1$ integers. The i-th integer should represent the maximum possible value of the minimum MEX when choosing $k=i+1$ nodes.

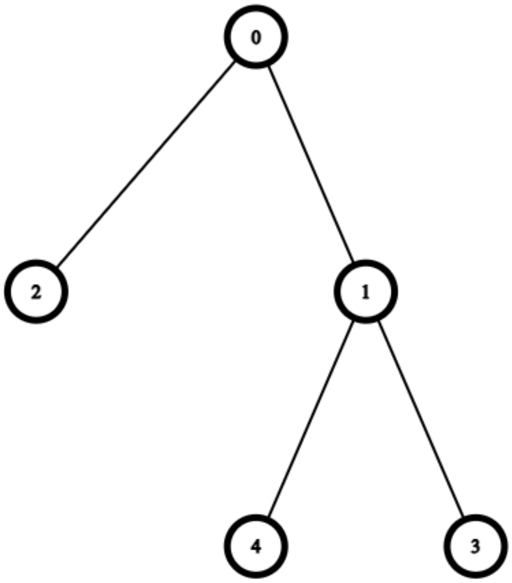
input

```
1
5
0 1
1 3
1 4
0 2
```

output

```
4 1 0 0
```

The given tree has $n=5$ nodes with the following structure:



- Node 0 is connected to nodes 1 and 2.
- Node 1 is connected to nodes 3 and 4.

For different values of k:

- For $k=2$, selecting nodes $\{2,3\}$ maximizes $f(S)$, yielding $\mathrm{MEX}(\mathcal{P}_{2,3})=4$.
- For $k=3$, selecting nodes $\{0,1,2\}$ yields a maximum $f(S)=1$.
- For $k=4$ and $k=5$, the maximum value of $f(S)$ is 0.