

Problem A. Zigzag Parity

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: White

You are given an integer n .

Construct a permutation p of length n such that for each i ($1 \leq i \leq n-2$), $(a_i + a_{i+1}) \bmod 2 \neq (a_{i+1} + a_{i+2}) \bmod 2$.

It is guaranteed that the answer always exists.

Input

The first line contains a single integer tc ($1 \leq tc < 1000$) — the number of testcases.

The only line of each testcase contains a single integer n ($1 \leq n \leq 5 \cdot 10^5$).

It is guaranteed that the sum of n over all the testcases doesn't exceed $5 \cdot 10^5$.

Output

For each testcase, print a permutation p .

If there are multiple answers, print any.

Example

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

Problem B. CoCo Count

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: `Light Blue`

You are given two arrays a and b , each of size n .

Count the number of good subarrays $[l, r]$.

A subarray $[l, r]$ is considered good if the following two conditions hold:

- the length of the subarray is greater than or equal to 2.
- a_l exists in the set $b_l, b_{l+1}, \dots, b_r$.
- a_r doesn't exist in the set $b_l, b_{l+1}, \dots, b_r$.

Input

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

The first line of each test case contains one single integer n ($1 \leq n \leq 5 \cdot 10^5$) — the length of the array.

The second line contains n integers a_i ($1 \leq a_i \leq 10^9$) — the array a .

The third line contains n integers b_i ($1 \leq b_i \leq 10^9$) — the array b .

It is guaranteed that the sum of n over all testcases doesn't exceed $5 \cdot 10^5$.

Output

For each test case print the number of good subarrays.

Example

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

Problem C. DGeneral Hamilton's Cubes

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: `Blue`

DGeneral Hamilton has n^3 small cubes of dimensions $1 \times 1 \times 1$. Each small cube has six faces numbered with the integers from 1 to 6, with each number appearing exactly once. The arrangement of these numbers on the faces is the same for all cubes (the small cubes are identical).

DGeneral Hamilton wants to build a large cube of dimensions $n \times n \times n$ using these small cubes. The large cube will also have 6 faces, numbered from 1 to 6, with each number appearing exactly once. Each face will consist of $n \times n$ visible squares, where each visible square shows the number on the corresponding face of the small cube placed there.

DGeneral Hamilton wants to know, how many different large cubes can he form?

Two large cubes are considered different if there is at least one face and at least one visible square on that face where the number on that square is different between the two cubes. Remember that the faces are numbered.

Print the answer modulo $10^9 + 7$.

Input

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

Each of the next t lines contains one integer n ($1 \leq n \leq 10^9$) — the dimension of the large cube DGeneral Hamilton wants to build. He has exactly n^3 small cubes.

Output

For each test case, print a single integer — the number of different large cubes that can be formed modulo $10^9 + 7$.

Example

standard input	standard output
2	24
1	75313406
2	

Problem D. Toward Divisibility

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: Gold

You are given an array a of n integers, you can do this operation **at most** once:

- Choose a subset of elements of the array and an integer and add this integer to all elements in this subset. In other words, choose a subset $S \subseteq \{1, 2, \dots, n\}$ and an integer X and set $a_i = a_i + X$ for each $i \in S$.

Print the minimum size of a subset you can choose such that after performing the operation, the GCD (Greatest Common Divisor) of all elements of the array is greater than one. If the GCD is already greater than one without performing the operation, print 0.

Input

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

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

Output

Print a single integer — the minimum size of the chosen subset or 0 if the GCD is already greater than one.

Example

standard input	standard output
5 1 2 3 4 5	3

Problem E. Permutation XORpectation

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: Lime Green

We define the score of a permutation p of length n as following :

$$\sum_{i=2}^n p_i \oplus p_{i-1}$$

Given n , find the expected value of the score of a random permutation of length n , modulo $10^9 + 7$.

Formally, let $M = 10^9 + 7$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \pmod{M}$. In other words, output such an integer x that $0 \leq x < M$ and $x \cdot q \equiv p \pmod{M}$.

Input

The first line contains a single integer tc : ($1 \leq tc \leq 10^5$) — the number of testcases.

The only line of each testcase contains a single integer n ($1 \leq n \leq 10^9$).

Output

For each testcase, print the expected value of the score of a random permutation of length n , modulo $10^9 + 7$.

Example

standard input	standard output
4	0
1	3
2	12
4	476632137
882	

Problem F. A bitty problem

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: Red

Ahmad is always tired.

He is so tired that he cannot make his small array beautiful :(

And by beautiful we mean that for $(2 \leq i \leq n)$ $a_i \oplus a_{i-1} = X$ for some constant X .

By $\oplus$ we mean the well known xor operation.

You will be given Ahmad's array, what is the minimum number of bits you need to change so that Ahmad's array is beautiful?

Input

The first line contains a single integer tc ($1 \leq tc \leq 1000$) — the number of testcases.

The first line of each test case contains a single integer n ($3 \leq n \leq 1000$).

The next line contains n integers a_i ($1 \leq a_i \leq 1000$).

Output

For each test case print one integer, the answer to the problem.

Example

standard input	standard output
5	4
5	0
1 2 3 4 5	0
3	7
1 2 1	4
3	
1 1 1	
4	
100 200 300 400	
3	
100 1 1	

Note

in the first test case we change 4 bits and the array becomes $[1, 2, 1, 2, 1]$ and xor of any two adjacent numbers is 3.

Problem G. Count the squares

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: Black

You are given n squares, where the side length of the i -th square is a_i .

Your task is to count the number of triples (i, j, k) where $(1 \leq i < j < k \leq n)$ such that the three squares at indices i, j , and k can be arranged to form a rectangle without any gaps or overlaps.

A square is considered a special case of a rectangle.

Input

The first line contains an integer t ($1 \leq t \leq 2 \cdot 10^5$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the number of squares.

The second line contains n space-separated integers a_i ($1 \leq a_i \leq 10^9$) — the side length of each square.

It is guaranteed that the sum of n over all test cases will not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer — the number of triples of squares that can form a rectangle.

Example

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

Note

In the first test case, the possible triples are: $(1, 2, 3)$, $(1, 2, 4)$, $(1, 3, 4)$, $(2, 3, 4)$.

Problem H. String Partition

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: Green

Given a string s of length n consisting of lowercase English letters.

The concatenation of a sequence of strings is the string that results from writing down these strings in the order they appear in the sequence. For example, the concatenation of the sequence ["code", "force", "s"] is the string "codeforces".

A partition of the string s is a sequence a of strings such that if we concatenate the strings in the sequence, the result will be the string s . Note that there may exist more than one possible partition of the string.

We call a partition of the string s good if the following holds:

- We say that a string b is considered **x-good** (for some positive integer x) if for every English letter that **appears in the string** b , the number of appearances of this letter in b is exactly x .
- A partition is considered good if there exists an integer x such that **all strings in the partition** are x -good.

The size of the partition is defined as the number of strings in that partition. Your task is to find the minimum size of a good partition.

Input

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

Each line of the next t lines contains a string s of length $|s|$ ($1 \leq |s| \leq 10^5$) — representing the string that is required to partition.

It's guaranteed that the sum of $|s|$ over all testcases doesn't exceed 10^5 .

Output

For each testcase, print a new line containing a single integer representing the minimum size of a good partition.

Example

standard input	standard output
3	2
codeforces	2
acpc	6
aywwaaaaa	

Problem I. Binary Reverser

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: Yellow

You are given an array a of n integers, and a binary string b of length n .

For each i from 1 to n (in order), you will perform one of the following actions:

- if b_i is '0', do nothing
- if b_i is '1', reverse the prefix of length i , that is for each integer j ($1 \leq j \leq i$), set $a_j = a_{i-j+1}$.

Your task is to print the final array after performing all the operations.

Input

The first line of the input contains a single integer tc ($1 \leq tc \leq 10^5$) — the number of testcases.

The first line of each test case contains a single integer n ($1 \leq n \leq 5 \cdot 10^5$) — the length of the array.

The second line of each test case contains n integers a_i ($1 \leq a_i \leq n$) — the array a .

The third line of each test case contains a binary string b of length n .

It is guaranteed that the sum of n over all testcases doesn't exceed $5 \cdot 10^5$.

Output

For each test case, print the final array after performing all the operations.

Example

standard input	standard output
2	3 1 2
3	2 1 5 4 1
1 2 3	
111	
5	
4 5 1 1 2	
01101	

Problem J. Down the rabbit hole we go!

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes
Balloon Color: `Silver`

Ahmad is very tired, so he decided to play Minecraft with his friend Apraham.

As soon as he entered the game, a guy named Steve gave him 2 fully grown rabbits and disappeared.

Apraham, curious about how rabbits live in this world, made some observations:

- Every X days, a pair of fully grown rabbits spawns 8 new baby rabbits.
- These newly spawned baby rabbits are smaller and cannot spawn new rabbits.
- After Y days, a baby rabbit grows up and becomes a fully grown rabbit. Once mature, it behaves like its parents and can start spawning new rabbits every X days.

Ahmad and Apraham now want to crash the server with these super cute rabbits UWU HeHe..

Today Ahmad has 2 fully grown rabbits and he is so curious about how many rabbits will he have after n days.

As i said Ahmad is STILL tired so this mission is yours obviously :)

Since the final answer could be astronomically large, please output the result modulo $10^9 + 7$.

Will you go down the rabbit hole?

Input

The only line contains three integers n, X, Y ($1 \leq n \leq 10^{18}$) ($1 \leq X, Y \leq 1000$)

Output

Print a single integer , the number of rabbits Ahmad has after n days mod $10^9 + 7$

Examples

standard input	standard output
2 2 3	10
20 2 3	8018
1000 4 5	641939610

Note

in the first test case:

after two days Ahmad will get extra 8 rabbits so $2+8=10$

Problem K. Least Common Route

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: **Purple**

You are given a tree of n nodes. Each node i has a value a_i . Count the number of simple paths where the LCM of the values of the nodes on the path equals X . In other words, count the number of pairs of nodes u, v ($1 \leq u \leq v \leq n$) where the LCM(Least Common Multiple) of the values a_i of the nodes on the simple path between u and v is equal to X .

Input

The first line contains two integers n, X ($1 \leq n \leq 10^5$)($1 \leq X \leq 10^6$).

The second line contains n integers a_i ($1 \leq a_i \leq 10^6$) — the values of the nodes.

Each of the following $n - 1$ lines contains two integers u, v ($1 \leq u, v \leq n$)($u \neq v$) — the edges of the tree.

It is guaranteed that the edges form a tree.

Output

Print a single integer — the number of paths with LCM equal to X .

Example

standard input	standard output
7 12 1 5 2 12 4 3 6 1 2 5 1 1 3 4 2 7 3 6 3	3

Problem L. Integer Average

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: Orange

Given an array a of length n consisting of positive integers.

The average of a sequence b of length m is equal to $\frac{b_1 + b_2 + \dots + b_m}{m}$. Note that the average is not necessarily an integer.

A sequence c is considered a subsequence of a if we can obtain c by removing some elements (possibly zero) from a (not necessarily adjacent elements). For example, $[1]$, $[1, 3]$ and $[1, 2, 4, 6, 1, 3]$ are all subsequences of $[1, 2, 3, 4, 5, 6, 9, 1, 3]$.

Your task is to find if there exists any subsequence of a consisting of **at least 2 elements**, such that the average of this subsequence is an integer.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 10^3$) — representing the number of testcases.

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

The second line of each testcase contains n space-separated integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — representing the elements of the array.

It's guaranteed the sum of n over all testcases doesn't exceed 10^5 .

Output

For each testcase, print a new line containing a single word — **YES** if the answer exists and **NO** otherwise. The checker is not case-sensitive, meaning that responses like **Yes**, **yES** and **yes** will be considered positive responses and responses like **No**, **nO** and **no** will be treated as negative responses.

Example

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

Problem M. Roots of Exclusion

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
Balloon Color: Pink

You are given an array of integers m of length n .

You have to construct a rooted tree of n nodes, such that for each node x , m_x is equal to the mex of the nodes in the subtree of x . In other words, if we take the set of nodes in the subtree of node x , the mex(minimum excluded value) of this set should be equal to m_x .

Note that the tree should be 0-indexed, the nodes should be numbered from 0 to $n - 1$.

Input

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

The first line of each testcase contains a single integer n ($1 \leq n \leq 10^5$).

The second line of each testcase consists of n integers m_i ($0 \leq m_i \leq n$).

It is guaranteed that there exists at least one solution for each testcase.

It is guaranteed that the sum of n overall testcases doesn't exceed 10^5 .

Output

For each testcase, print n lines.

On the first line print a single integer $root$ ($0 \leq root \leq n - 1$) — the root of the tree.

On each of the next $n - 1$ lines, print two integers u, v ($0 \leq u, v \leq n - 1$) — the edges of the tree.

If there are many answers, print any.

Example

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