Problem A. Goals, Goals! Everywhere

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Balloon Color:	`Purple`

Ammar's football team consists of n players. The last match was incredible, and player i contributed to scoring a_i goals.

A contribution is either scoring a goal or assisting in scoring. A goal is scored by exactly one player, and at most one player can assist in scoring it. A player can't score and assist in scoring the same goal.

Determine the minimum and maximum number of goals that can be scored.

Input

The first line contains one integer t $(1 \leq t \leq 10^3)$ — the number of test cases.

The first line of each test case consists of a single integer n $(1 \leq n \leq 3 \cdot 10^5)$ — the number of players in Ammar's team.

The second line of each testcase consists of n integers a_i $(0 \leq a_i \leq 10^9)$ — the contribution of each player.

It is guaranteed that the sum of n over all test cases doesn't exceed $3 \cdot 10^5$.

Output

For each testcase, output two integers: the minimum and maximum number of goals that can be scored.

Example

standard input	standard output
`3`	`7 13`
`3`	`10 10`
`5 2 6`	`91 182`
`1`	
`10`	
`5`	
`59 35 12 64 12`	

Note

Consider the first testcase. The maximum number of goals is the sum of all contributions, which is 13.

The minimum number of goals is 7 goals. The third player scores 6 goals, each assisted by either the first or the second player, **not both**, leaving one contribution which is considered a goal.

Problem B. Depth Range Update

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Balloon Color:	`Red`

Given a tree with n nodes rooted at 1 and an array a where a_i is the value of the node i.

We define $depth_u$ as the number of edges on the shortest path from node u to node 1.

You are to process q queries of two types:

— $1\ l\ r\ x$: for every node u that satisfies $l \leq dpth_u \leq r$ do the following assignment: $a_u := a_u \oplus x$, it's guaranteed that $r - l = 1$ (where $\oplus$ corresponds to the XOR bitwise operation).

— $2\ u$: print the following sum : $\sum_{v \in subtree_u} xorpath(u, v)$, where $xorpath(u, v)$ is the xor sum of node values on the shortest path between u and v.

Input

The first line contains one integer t $(1 \leq t \leq 100)$ — the number of test cases.

The first line of each testcase contains two integers n and q $(2 \leq n \leq 10^5,\ 1 \leq q \leq 10^5)$

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(0 \leq a_i < 2^{20})$.

The next $n - 1$ lines contain the tree edges $u\ v$ $(1 \leq u, v \leq n)$.

The next q lines contain the queries described above $(0 \leq l \leq r \leq n), (r - l = 1)$, $(0 \leq x < 2^{20})$, $(1 \leq u \leq n)$.

It's guaranteed that the sum of n over all test cases doesn't exceed 10^5 and the sum of q over all test cases doesn't exceed 10^5.

Output

For each query of the second type, print the required sum.

Example

standard input	standard output

2
4 4
2 3 0 7
2 4
1 3
4 1
2 4
1 0 1 4
1 2 3 6
1 0 1 4
6 1
4 7 7 3 3 3
1 4
4 6
3 5
3 4
6 2
2 2
	11
	7

Problem C. Variety Hater

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Balloon Color:	`Orange`

Yazan has an array a consisting of n integers.

Yazan can perform the following operation at most k times:

- Choose an index i $(1 \le i \le n)$ and either increase a_i by one or decrease it by one. In other words, either apply $a_i := a_i - 1$ or $a_i := a_i + 1$.

Yazan hates variety, so he wants to minimize the number of distinct elements in the array. Can you help him find the minimum number of distinct elements he can have after performing no more than k operations optimally?

Input

The first line contains one integer t $(1 \le t \le 100)$ — the number of test cases.

The first line of each testcase contains two integers n and k $(1 \le n \le 300,\ 0 \le k \le 10^{12})$ — the number of elements and the number of available operations.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$.

It is guaranteed that the sum of n over all test cases doesn't exceed 300.

Output

For each test case, output the minimum number of distinct elements Yazan can have after performing no more than k operations.

Example

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

Problem D. Make It Minimum

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

ALI-SAIRO has a string s of length n containing only digit numbers from '0' to '9'.

Let's define d_i $(1 \leq i < n)$ as the number whose decimal representation is $s_i s_{i+1}$.

We define F as:

$$F(s) = \sum_{i=1}^{n-1} d_i$$

For example : $s =$ "0130" then $F = 01 + 13 + 30 = 44$.

In one operation, you can choose i and j $(1 \leq i \leq j \leq n)$; $(i,\ j$ **don't have to be adjacent**), and swap s_i and s_j.

Find the minimum number of operations required to minimize $F(s)$, and determine the minimum value of $F(s)$.

Input

The first line contains one integer t $(1 \leq t \leq 10^3)$ — the number of test cases.

The first line of each test case contains a single integer n : $(1 \leq n \leq 10^6)$ — length of the string.

The second line of each test case contains a string s of length n.

It is guaranteed that the sum of n over all test cases doesn't exceed 10^6.

Output

For each testcase, output the minimum number of operations required to minimize $F(s)$, and determine the minimum value of $F(s)$.

Example

standard input	standard output

3
2
21
8
99999999
1
9
	1 12
	0 693
	0 0

Problem E. Changes in Antwanland

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Balloon Color:	`Pink`

Antwanland is a tree of size n (a tree is a connected graph with no loops), and Antwan has a lucky number k.

Antwan wants to reduce the size of his land to k while ensuring it remains a connected tree.

The citizens of Antwanland are smart and always travel the shortest path between any two cities.

Antwan cares about his citizens and wants to choose k cities such that the maximum journey (number of cities) between any two cities is minimized.

Can you help Antwan achieve this?

Input

The first line contains two integers n and k ($1 \leq k \leq n \leq 10^3$) — the size of Antwanland and his lucky number.

Each of the next $n-1$ lines contains two integers v and u ($1 \leq v, u \leq n$) — the description of the edges of the tree. It's guaranteed that the given edges form a valid tree.

Output

A single integer, the minimum maximum journey between any two cities after helping Antwan.

Examples

standard input	standard output

4 3
1 2
2 3
3 4
	3

7 3
1 2
1 3
1 4
2 5
3 6
4 7
	3

Problem F. Tree XOR

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Balloon Color:	`Black`

As usual, SUL was thinking about many strange questions, such as did Newton discover that he could sit under trees, and that apples fall down on the same day he discovered gravity, and why was it one apple not k apples? So think about the following problem

You are given a tree of size n, each node x has a value a_x.

Find a connected component G in the tree such that the bitwise XOR $\oplus$ of a_x for all nodes $x \in G$ equals k, or determine that no such component exists.

Input

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

The first line of each test case consists of two integers n and k ($1 \leq n \leq 5 \cdot 10^4$, $0 \leq k \leq 63$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 63$).

Each of the next $n-1$ lines contains two integers v and u ($1 \leq v, u \leq n$) — the description of the edges of the tree. It's guaranteed that the given edges form a valid tree.

It is guaranteed that the sum of n over all test cases doesn't exceed $5 \cdot 10^4$.

Output

For each testcase, output "Yes" if there is a connected component that satisfies the above conditions, then output s, the size of the connected component, followed by s integers representing the nodes in G.

If no such connected component exists, output "No".

Example

standard input	standard output

2
3 0
1 13 3
1 2
2 3
10 5
7 23 3 1 2 3 6 10 2 4
1 2
2 3
2 4
3 5
4 6
1 7
2 8
6 9
6 10
	NO
	YES
	3
	6 9 10

Note

Consider the first testcase. An empty component is not a valid answer.

Problem G. The Elden Program

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Balloon Color:	`Light Purple`

Zack is a professional gamer who loves to play challenging games. Recently, he has been playing a very difficult game called The Elden Program. In The Elden Program, there are n monsters standing in a row, where each monster i has an energy level of a_i.

Fortunately, Zack found a powerful spell that he can use against the monsters. This spell can be cast on a single monster. When he casts this spell on a monster x, monster x becomes fixed in place, while every other monster will:

- Pick a direction, either to the left or to the right, and never change it.
- Start moving in the chosen direction simultaneously, each second moving to an adjacent index. Each move takes one unit of time.
- If two different monsters i and j encounter each other, and $a_i > a_j$, then monster i will **defeat** monster j and the energy of monster i will increase by a_j.
- If two different monsters i and j encounter each other, and $a_i = a_j$, then both monsters will **vanish**.

Keep in mind that when a monster vanishes, that doesn't mean they are defeated.

The rest of the monsters perform those actions in such a way to minimize the time it takes to **defeat** monster x.

Zack wants to plan out his assault on the monsters, so he asks you to determine the time it takes to defeat each monster if the spell was cast on it, or determine that it's impossible to defeat it.

Input

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

The first line of each test case consists of a single integer n ($1 \le n \le 3 \cdot 10^5$) — the number of monsters.

The second line of each testcase consists of n integers ($1 \le a_i \le 10^9$) — denoting the energy of each monster.

It is guaranteed that the sum of n over all test cases doesn't exceed $3 \cdot 10^5$.

Output

For each testcase, output n numbers, the time it takes to defeat monster i if the spell was cast on it, or "-1" if it is impossible to **defeat** it.

Example

standard input	standard output
6	1 3 1 1 3 2 1 3 1 1 3 1
12	1 3 1 1 3 2 1 6 1 2 1 2
5 6 3 1 5 5 5 8 1 6 7 6	1 3 1 1 3 2 1 -1 1 2 1 2
12	1 3 1 1 3 2 1 3 1 1 3 1
5 6 3 1 5 5 5 8 1 6 1 5	-1 -1 -1 -1 -1
12	-1 1 2 1 4
5 6 3 1 5 5 5 100 1 6 1 5	
12	
5 6 3 1 5 5 5 8 1 6 7 6	
5	
1 1 1 1 1	
5	
10 1 1 1 2	

Note

Consider the second testcase. When you cast a spell on the 8-th monster $a_8 = 8$, all the other monsters will work together to **defeat** it. The monsters on the right of the 8-th monster will go to the right so they won't increase it's power. The 2-nd monster $a_2 = 6$ will go towards the 8-th monster while the other monsters in between the two monsters will go into the 2-nd monster increasing it's power.

Problem H. Good Array

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Balloon Color:	`Silver`

Majed found an array a consisting of n integers.

He can do the following operation many (possibly zero) times:

- Choose two indexes i and j such that a_i is even and perform $a_i = \frac{a_i}{2}$ and $a_j = a_j \cdot 2$.

Majed considers an integer x **good** if there are at least x elements in a which are greater than or equal to x.

Majed wants to determine the maximum possible integer x that is considered good after performing the operation several (possibly zero) times.

Input

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

The first line of each test case consists of a single integer n $(1 \le n \le 2 \cdot 10^5)$.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$.

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

Output

For each test case, output the maximum possible integer x that is considered good after performing the operation several (possibly zero) times.

Example

standard input	standard output

3
5
1 2 8 3 1
3
1 1 1
5
1 3 5 99 7
	3
	1
	3

Note

Consider the first testcase. Majed can choose $i = 3$ and $j = 2$, making $a = [1, 4, 4, 3, 1]$, which has 3 elements greater than or equal to 3.

Problem I. Homies and Not Homies

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Balloon Color:	`Dark Blue`

Homies and Not Homies are rivals from the start of time. Homies recent project involved bulbs and lights. They bought a bunch of bulbs and arranged them in rows. When the electricity is turned on, row i represents the number i in binary representation (without leading zeros).

For example, if $n = 3$, the bulbs will be arranged in the following way:

- the first row will be: 1
- the second row will be: 10
- the third row will be: 11

where 0 represents an off-bulb and 1 represents an on-bulb.

Not Homies, Homies archenemy, have vandalized the light bulbs. But due to their limited brain function, they can only comprehend even numbers, so they only destroyed bulbs at even indexes (starting from left to right). When the Homies saw this, they decided to make fun of them by counting the number of light bulbs that are still turned on.

Input

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

The first line of each test case consists of a single integer n $(1 \le n \le 10^9)$ — the number of rows.

Output

Output the number of bulbs that are still on after Not Homies messed with the bulbs.

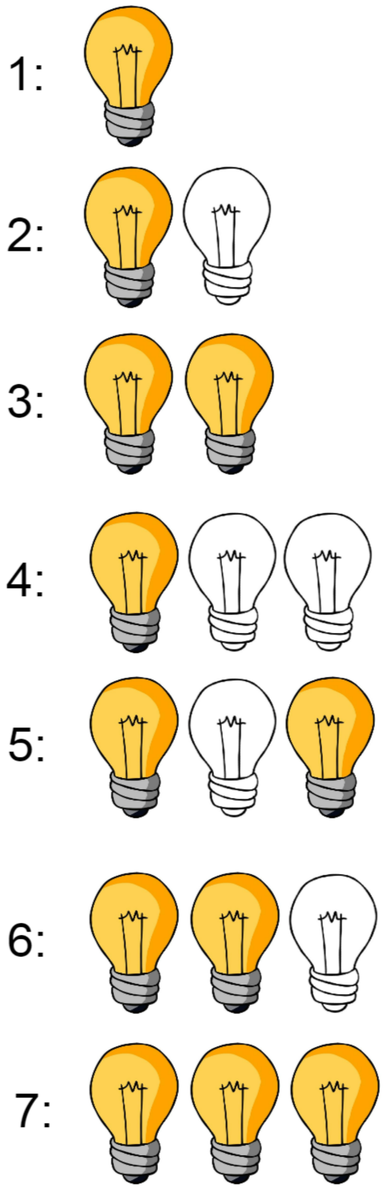
Example

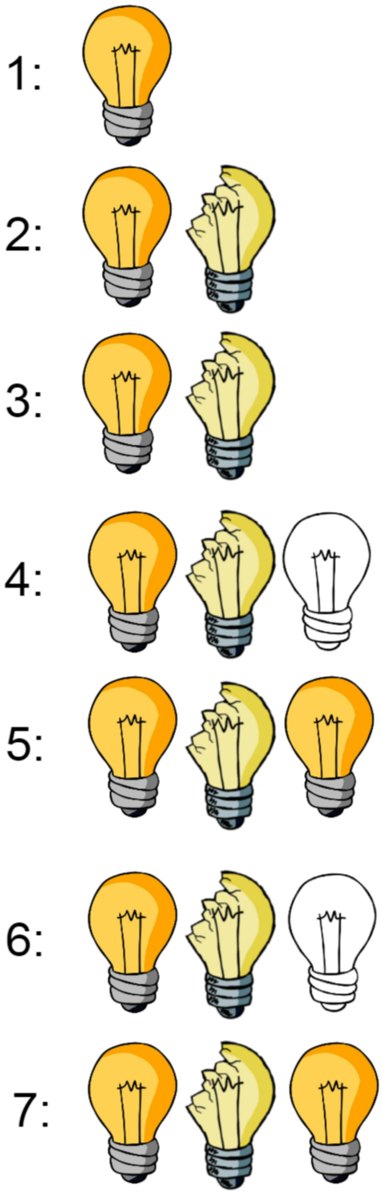
standard input	standard output
3	6
5	9
7	13
10	

Note

Consider the second testcase:

Those are the light bulbs before and after Not Homies messed with them, as you can see **the parity of the light bulbs** is relative to the largest active light bulb.


1:
2:
3:
4:
5:
6:
7:


1:
2:
3:
4:
5:
6:
7:

Problem J. Game of Primes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Balloon Color:	`Gold`

Ammar and Antwan are playing a game with two multisets a and b, initially a contains n integers and b is empty.

Define $F(x)$ as the greatest divisor of x less than x, in other words, the greatest y such that $y < x$ and y divides x .

The game proceeds as follows: the players take turns with Ammar starting. On each turn, a player will perform the following operation:

- Choose an integer x from multiset a, remove it and either add $F(x)$ or $\frac{x}{F(x)}$ to b.

Ammar loves primes and wants to maximize the number of primes in b, while Antwan hates primes and wants to minimize this number. In their universe, the number 1 is considered a prime number.

Determine the number of primes (considering 1 a prime) in b after the game ends if both players play optimally.

Input

The first line contains one integer t ($1 \le t \le 10^3$) — the number of test cases.

The first line of each test case consists of a single integer n ($1 \le n \le 10^5$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($2 \le a_i \le 10^6$).

It is guaranteed that the sum of n over all test cases doesn't exceed 10^5.

Output

For each test case, output the number of primes (considering 1 a prime) in b after the game ends if both players play optimally.

Example

standard input	standard output

2
5
2 3 20 30 5
2
10 5
	4
	2

Problem K. Minimum Sum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Balloon Color:	`White`

Given n n-digit numbers (there may be leading zeros).

For each i **in increasing order** $(1 \leq i \leq n)$, you **must** choose $j \neq i$ and perform the following operation:

- $a_i := a_j$ (assign the j-th number to the i-th number).

Find the minimum sum of all n numbers modulo 998244353 after performing all the operations (you need to find the minimum sum first, then print it modulo 998244353).

Input

The first line consists of one integer n $(2 \leq n \leq 700)$.

The following n lines consist of the numbers a_i (where a_i has exactly n digits).

Output

The only line consists of one integer, which is the minimum sum modulo 998244353 after performing all the operations.

Examples

standard input	standard output
4 3142 5310 0341 3423	1364
3 032 102 999	306

Problem L. The Shrine of the Father of Forces

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Balloon Color:	`Green`

The Shrine of the Father of Forces is a renowned shrine known for its miraculous ability to settle any dispute between two parties. The shrine resolves disputes by asking each party a series of questions, and the party that answers correctly wins the dispute.

Mohanad finds this fascinating and realizes that if he knows the questions in advance, he can always win. Fortunately, Mohanad has discovered the type of questions the shrine will ask. The shrine will ask t questions, each of which is:

Given an odd integer n, find the number of permutations of size n that satisfy the following conditions:

- The values at odd indices are first increasing and then decreasing (they can be just increasing or just decreasing).
- The values at even indices are less than their neighboring values at odd indices (the first and last indices have only one neighboring value).

Please note that the permutation is 0-indexed

A permutation of n integers is an array containing all numbers from 1 to n exactly once. For example, the arrays $[1]$, $[2, 1, 3]$, $[5, 4, 3, 2, 1]$ are permutations, while the arrays $[1, 1]$, $[100]$, $[1, 2, 4, 5]$ are not.

Knowing the questions beforehand, Mohanad asked you to help him solve them.

Input

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

The first line of each test case consists of a single integer n $(1 \le n < 10^5)$.

It is guaranteed that the sum of n over all test cases doesn't exceed 10^5, and n is an odd integer.

Output

For each testcase, output the number of permutations that satisfy the above conditions, taken modulo $10^9 + 7$.

Example

standard input	standard output
5 1 3 5 7 9	1 2 16 176 2560

Problem M. Kaaa

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Balloon Color:	`Lime`

Mohanad recently fixed his sleeping schedule (he has been trying to for the past four years!). A huge reason for that fix was the annoying crow that would scream "KaaaKaaaKaaa"(but in Arabic) at exactly 8am.

To honor that crow, Mohanad decided to create a problem about it. You are given what Mohanad heard at exactly 8am, and you need to determine whether he woke up or stayed asleep.

Input

The first line contains one string s ($1 \leq |s| \leq 100$) — what Mohanad heard at 8am.

Output

If Mohanad heard "KaaaKaaaKaaa", output "Woken Up"; otherwise, output "Still Asleep".

Examples

standard input	standard output
`KaaaKaaaKaaa`	`Woken Up`
`KaaaKaaKa`	`Still Asleep`
`wake_up_its_time_for_school`	`Still Asleep`
`wake_up_you_got_a_contest_to_judge`	`Still Asleep`