A. Filling Diamonds

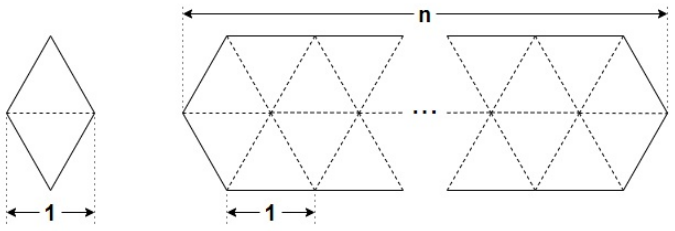
time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have integer n. Calculate how many ways are there to fully cover belt-like area of $4n-2$ triangles with diamond shapes.

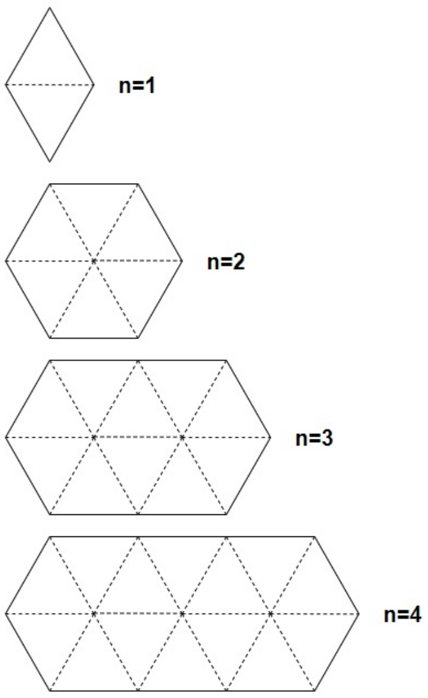
Diamond shape consists of two triangles. You can move, rotate or flip the shape, but you cannot scale it.

2 coverings are different if some 2 triangles are covered by the same diamond shape in one of them and by different diamond shapes in the other one.

Please look at pictures below for better understanding.



On the left you can see the diamond shape you will use, and on the right you can see the area you want to fill.



These are the figures of the area you want to fill for $n = 1, 2, 3, 4$.

You have to answer t independent test cases.

Input

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

Each of the next t lines contains a single integer n ($1 \le n \le 10^9$).

Output

For each test case, print the number of ways to fully cover belt-like area of $4n-2$ triangles using diamond shape. It can be shown that under given constraints this number of ways doesn't exceed 10^{18}.

Example

input

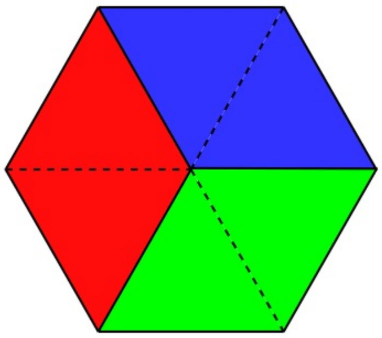
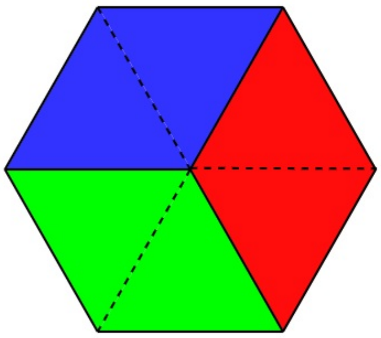
```
2
2
1
```

output

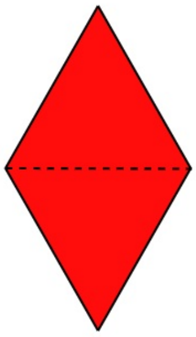
```
2
1
```

Note

In the first test case, there are the following 2 ways to fill the area:

In the second test case, there is a unique way to fill the area:



B. Sorted Adjacent Differences

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have array of n numbers $a_1, a_2, \ldots, a_n$.

Rearrange these numbers to satisfy $|a_1 - a_2| \le |a_2 - a_3| \le \ldots \le |a_{n-1} - a_n|$, where $|x|$ denotes absolute value of x. It's always possible to find such rearrangement.

Note that all numbers in a are not necessarily different. In other words, some numbers of a may be same.

You have to answer independent t test cases.

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 single integer n ($3 \le n \le 10^5$) — the length of array a. It is guaranteed that the sum of values of n over all test cases in the input does not exceed 10^5.

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

Output

For each test case, print the rearranged version of array a which satisfies given condition. If there are multiple valid rearrangements, print any of them.

Example

input

```
2
6
5 -2 4 8 6 5
4
8 1 4 2
```

output

```
5 5 4 6 8 -2
1 2 4 8
```

Note

In the first test case, after given rearrangement, $|a_1 - a_2| = 0 \le |a_2 - a_3| = 1 \le |a_3 - a_4| = 2 \le |a_4 - a_5| = 2 \le |a_5 - a_6| = 10$. There are other possible answers like "5 4 5 6 -2 8".

In the second test case, after given rearrangement, $|a_1 - a_2| = 1 \le |a_2 - a_3| = 2 \le |a_3 - a_4| = 4$. There are other possible answers like "2 4 8 1".

C. Powered Addition

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have an array a of length n. For every positive integer x you are going to perform the following operation during the x-th second:

- Select some distinct indices $i_1, i_2, \ldots, i_k$ which are between 1 and n inclusive, and add 2^{x-1} to each corresponding position of a. Formally, $a_{i_j} := a_{i_j} + 2^{x-1}$ for $j = 1, 2, \ldots, k$. **Note that you are allowed to not select any indices at all.**

You have to make a nondecreasing as fast as possible. Find the smallest number T such that you can make the array nondecreasing after at most T seconds.

Array a is nondecreasing if and only if $a_1 \le a_2 \le \ldots \le a_n$.

You have to answer t independent test cases.

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 single integer n ($1 \le n \le 10^5$) — the length of array a. It is guaranteed that the sum of values of n over all test cases in the input does not exceed 10^5.

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

Output

For each test case, print the minimum number of seconds in which you can make a nondecreasing.

Example

input

```
3
4
1 7 6 5
5
1 2 3 4 5
2
0 -4
```

output

```
2
0
3
```

Note

In the first test case, if you select indices $3, 4$ at the 1-st second and 4 at the 2-nd second, then a will become $[1, 7, 7, 8]$. There are some other possible ways to make a nondecreasing in 2 seconds, but you can't do it faster.

In the second test case, a is already nondecreasing, so answer is 0.

In the third test case, if you do nothing at first 2 seconds and select index 2 at the 3-rd second, a will become $[0, 0]$.

D. Edge Weight Assignment

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

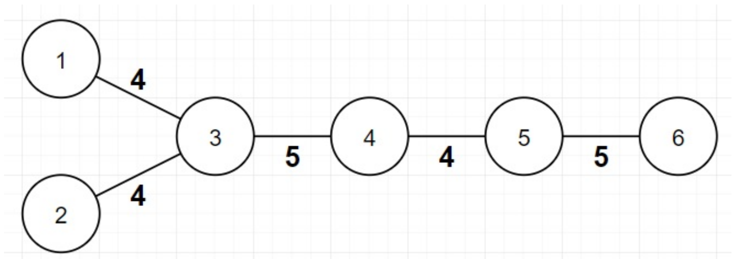
output: standard output

You have unweighted tree of n vertices. You have to assign a **positive** weight to each edge so that the following condition would hold:

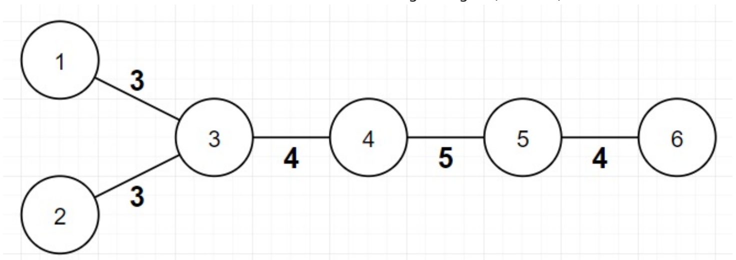
- For every two different leaves v_1 and v_2 of this tree, bitwise XOR of weights of all edges on the simple path between v_1 and v_2 has to be equal to 0.

Note that you can put **very large** positive integers (like $10^{(10^{10})}$).

It's guaranteed that such assignment always exists under given constraints. Now let's define f as **the number of distinct weights** in assignment.



In this example, assignment is valid, because bitwise XOR of all edge weights between every pair of leaves is 0. f value is 2 here, because there are 2 distinct edge weights(4 and 5).



In this example, assignment is invalid, because bitwise XOR of all edge weights between vertex 1 and vertex 6 ($3, 4, 5, 4$) is not 0.

What are the minimum and the maximum possible values of f for the given tree? Find and print both.

Input

The first line contains integer n ($3 \le n \le 10^5$) — the number of vertices in given tree.

The i-th of the next $n - 1$ lines contains two integers a_i and b_i ($1 \le a_i < b_i \le n$) — it means there is an edge between a_i and b_i. It is guaranteed that given graph forms tree of n vertices.

Output

Print two integers — the minimum and maximum possible value of f can be made from valid assignment of given tree. Note that it's always possible to make an assignment under given constraints.

Examples

input

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

output

```
1 4
```

input

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

output

```
3 3
```

input

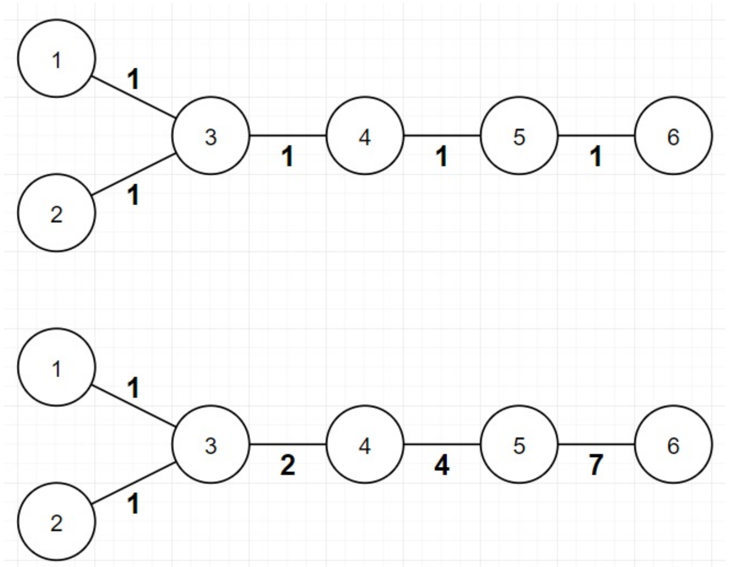
```
7
1 2
2 7
3 4
4 7
5 6
6 7
```

output

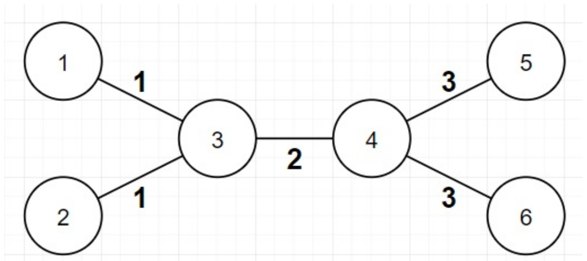
```
1 6
```

Note

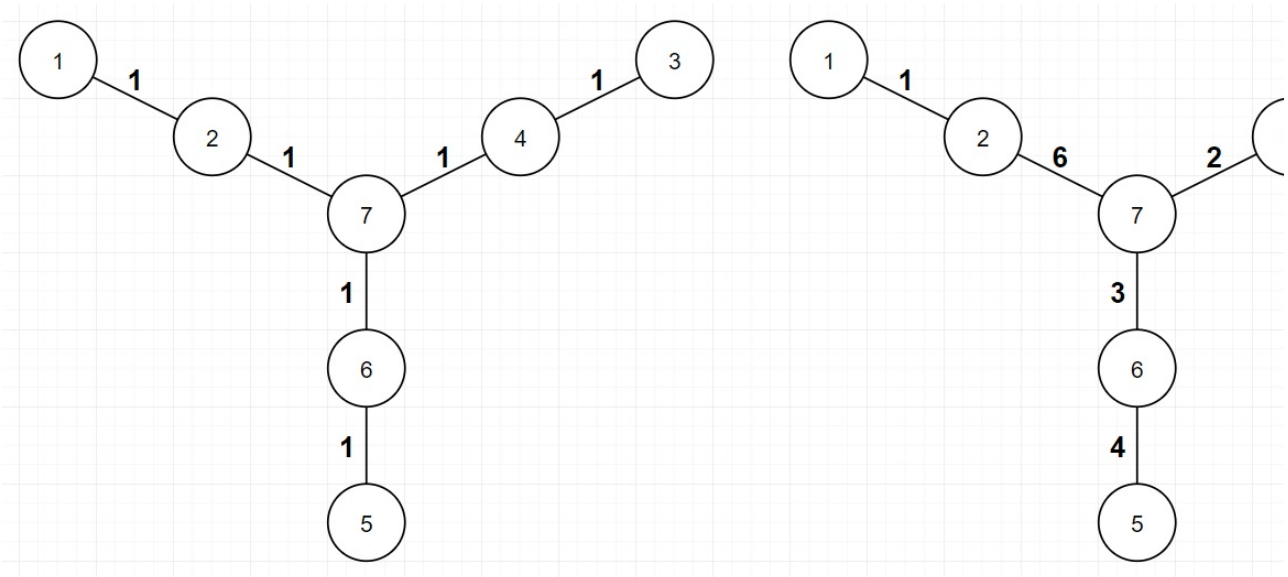
In the first example, possible assignments for each minimum and maximum are described in picture below. Of course, there are multiple possible assignments for each minimum and maximum.



In the second example, possible assignments for each minimum and maximum are described in picture below. The f value of valid assignment of this tree is always 3.



In the third example, possible assignments for each minimum and maximum are described in picture below. Of course, there are multiple possible assignments for each minimum and maximum.



E. Perfect Triples

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Consider the infinite sequence s of positive integers, created by repeating the following steps:

1. Find the lexicographically smallest triple of positive integers (a, b, c) such that

    - $a \oplus b \oplus c = 0$, where $\oplus$ denotes the bitwise XOR operation.
    - a, b, c are not in s.

    Here triple of integers (a_1, b_1, c_1) is considered to be lexicographically smaller than triple (a_2, b_2, c_2) if sequence $[a_1, b_1, c_1]$ is lexicographically smaller than sequence $[a_2, b_2, c_2]$.
2. Append a, b, c to s in this order.
3. Go back to the first step.

You have integer n. Find the n-th element of s.

You have to answer t independent test cases.

A sequence a is lexicographically smaller than a sequence b if in the first position where a and b differ, the sequence a has a smaller element than the corresponding element in b.

Input

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

Each of the next t lines contains a single integer n ($1 \le n \le 10^{16}$) — the position of the element you want to know.

Output

In each of the t lines, output the answer to the corresponding test case.

Example

input

```
9
1
2
3
4
5
6
7
8
9
```

output

```
1
2
3
4
8
12
5
10
15
```

Note

The first elements of s are $1, 2, 3, 4, 8, 12, 5, 10, 15, \dots$