A. Add and Divide

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

You have two positive integers a and b.

You can perform two kinds of operations:

- $a = \lfloor \frac{a}{b} \rfloor$ (replace a with the integer part of the division between a and b)
- $b = b + 1$ (increase b by 1)

Find the minimum number of operations required to make $a = 0$.

Input

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

The only line of the description of each test case contains two integers a, b ($1 \le a, b \le 10^9$).

Output

For each test case, print a single integer: the minimum number of operations required to make $a = 0$.

Example

input

```
6
9 2
1337 1
1 1
50000000 4
991026972 997
1234 5678
```

output

```
4
9
2
12
3
1
```

Note

In the first test case, one of the optimal solutions is:

1. Divide a by b. After this operation $a = 4$ and $b = 2$.
2. Divide a by b. After this operation $a = 2$ and $b = 2$.
3. Increase b. After this operation $a = 2$ and $b = 3$.
4. Divide a by b. After this operation $a = 0$ and $b = 3$.

B. Replace and Keep Sorted

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given a positive integer k, two arrays are called k-similar if:

- they are **strictly increasing**;
- they have the same length;
- all their elements are positive integers between 1 and k (inclusive);
- they differ in **exactly** one position.

You are given an integer k, a **strictly increasing** array a and q queries. For each query, you are given two integers $l_i \le r_i$. Your task is to find how many arrays b exist, such that b is k-similar to array $[a_{l_i}, a_{l_i+1} \ldots, a_{r_i}]$.

Input

The first line contains three integers n, q and k ($1 \le n, q \le 10^5$, $n \le k \le 10^9$) — the length of array a, the number of queries and number k.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le k$). This array is strictly increasing — $a_1 < a_2 < \ldots < a_n$.

Each of the following q lines contains two integers l_i, r_i ($1 \le l_i \le r_i \le n$).

Output

Print q lines. The i-th of them should contain the answer to the i-th query.

Examples

input

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

output

```
4
3
```

input

```
6 5 10
2 4 6 7 8 9
1 4
1 2
3 5
1 6
5 5
```

output

```
8
9
7
6
9
```

Note

In the first example:

In the first query there are 4 arrays that are 5-similar to $[2,4]$: $[1,4], [3,4], [2,3], [2,5]$.

In the second query there are 3 arrays that are 5-similar to $[4,5]$: $[1,5], [2,5], [3,5]$.

C. Floor and Mod

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A pair of positive integers (a, b) is called **special** if $\lfloor \frac{a}{b} \rfloor = a \bmod b$. Here, $\lfloor \frac{a}{b} \rfloor$ is the result of the integer division between a and b, while $a \bmod b$ is its remainder.

You are given two integers x and y. Find the number of special pairs (a, b) such that $1 \le a \le x$ and $1 \le b \le y$.

Input

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

The only line of the description of each test case contains two integers x, y ($1 \le x, y \le 10^9$).

Output

For each test case print the answer on a single line.

Example

input

```
9
3 4
2 100
4 3
50 3
12 4
69 420
12345 6789
123456 789
12345678 9
```

output

```
1
0
2
```

```
3
5
141
53384
160909
36
```

Note

In the first test case, the only special pair is $(3, 2)$.

In the second test case, there are no special pairs.

In the third test case, there are two special pairs: $(3, 2)$ and $(4, 3)$.

D. Multiples and Power Differences

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a matrix a consisting of positive integers. It has n rows and m columns.

Construct a matrix b consisting of positive integers. It should have the same size as a, and the following conditions should be met:

- $1 \le b_{i,j} \le 10^6$;
- $b_{i,j}$ is a multiple of $a_{i,j}$;
- the absolute value of the difference between numbers in any adjacent pair of cells (two cells that share the same side) in b is equal to k^4 for some integer $k \ge 1$ (k is not necessarily the same for all pairs, it is own for each pair).

We can show that the answer always exists.

Input

The first line contains two integers n and m ($2 \le n, m \le 500$).

Each of the following n lines contains m integers. The j-th integer in the i-th line is $a_{i,j}$ ($1 \le a_{i,j} \le 16$).

Output

The output should contain n lines each containing m integers. The j-th integer in the i-th line should be $b_{i,j}$.

Examples

input

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

output

```
1 2
2 3
```

input

```
2 3
16 16 16
16 16 16
```

output

```
16 32 48
32 48 64
```

input

```
2 2
3 11
12 8
```

output

```
327 583
408 664
```

Note

In the first example, the matrix a can be used as the matrix b, because the absolute value of the difference between numbers in any adjacent pair of cells is $1 = 1^4$.

In the third example:

- 327 is a multiple of 3, 583 is a multiple of 11, 408 is a multiple of 12, 664 is a multiple of 8;
- $|408 - 327| = 3^4$, $|583 - 327| = 4^4$, $|664 - 408| = 4^4$, $|664 - 583| = 3^4$.

E. Move and Swap

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given $n-1$ integers $a_2, \ldots, a_n$ and a tree with n vertices rooted at vertex 1. The leaves are all at the same distance d from the root.

Recall that a tree is a connected undirected graph without cycles. The distance between two vertices is the number of edges on the simple path between them. All non-root vertices with degree 1 are leaves. If vertices s and f are connected by an edge and the distance of f from the root is greater than the distance of s from the root, then f is called a child of s.

Initially, there are a red coin and a blue coin on the vertex 1. Let r be the vertex where the red coin is and let b be the vertex where the blue coin is. You should make d moves. A move consists of three steps:

- Move the red coin to any child of r.
- Move the blue coin to any vertex b' such that $dist(1, b') = dist(1, b) + 1$. Here $dist(x, y)$ indicates the length of the simple path between x and y. Note that b and b' are not necessarily connected by an edge.
- You can optionally swap the two coins (or skip this step).

Note that r and b can be equal at any time, and there is no number written on the root.

After each move, you gain $|a_r - a_b|$ points. What's the maximum number of points you can gain after d moves?

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 a single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of vertices in the tree.

The second line of each test case contains $n-1$ integers $v_2, v_3, \ldots, v_n$ ($1 \le v_i \le n$, $v_i \ne i$) — the i-th of them indicates that there is an edge between vertices i and v_i. It is guaranteed, that these edges form a tree.

The third line of each test case contains $n-1$ integers $a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the numbers written on the vertices.

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

Output

For each test case, print a single integer: the maximum number of points you can gain after d moves.

Example

input

```
4
14
1 1 1 2 3 4 4 5 5 6 7 8 8
2 3 7 7 6 9 5 9 7 3 6 6 5
6
1 2 2 3 4
32 78 69 5 41
15
1 15 1 10 4 9 11 2 4 1 8 6 10 11
62 13 12 43 39 65 42 86 25 38 19 19 43 62
15
11 2 7 6 9 8 10 1 1 1 5 3 15 2
50 19 30 35 9 45 13 24 8 44 16 26 10 40
```

output

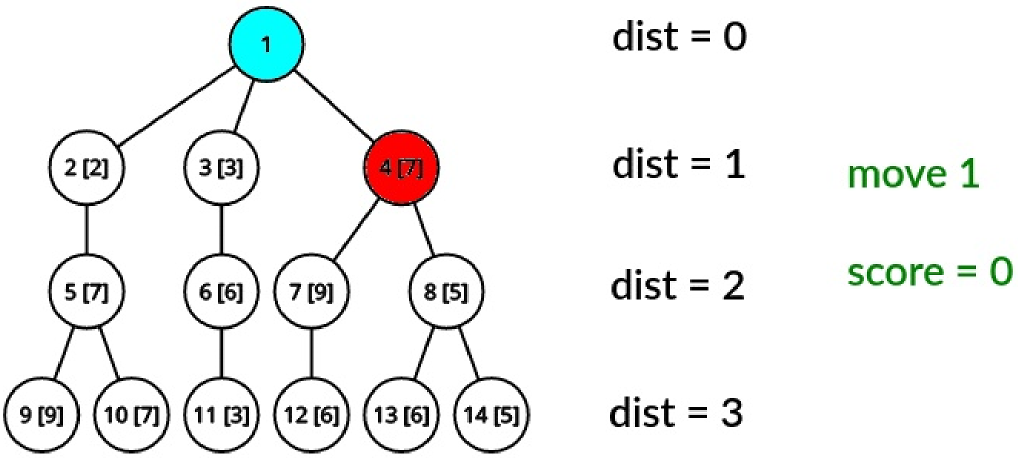
```
14
45
163
123
```

Note

In the first test case, an optimal solution is to:

- move 1: $r = 4$, $b = 2$; no swap;
- move 2: $r = 7$, $b = 6$; swap (after it $r = 6$, $b = 7$);
- move 3: $r = 11$, $b = 9$; no swap.

The total number of points is $|7 - 2| + |6 - 9| + |3 - 9| = 14$.



In the second test case, an optimal solution is to:

- move 1: $r = 2$, $b = 2$; no swap;
- move 2: $r = 3$, $b = 4$; no swap;
- move 3: $r = 5$, $b = 6$; no swap.

The total number of points is $|32 - 32| + |78 - 69| + |5 - 41| = 45$.

F. Copy or Prefix Sum

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array of integers $b_1, b_2, \ldots, b_n$.

An array $a_1, a_2, \ldots, a_n$ of integers is **hybrid** if for each i ($1 \le i \le n$) at least one of these conditions is true:

- $b_i = a_i$, or
- $b_i = \sum_{j=1}^{i} a_j$.

Find the number of hybrid arrays $a_1, a_2, \ldots, a_n$. As the result can be very large, you should print the answer modulo $10^9 + 7$.

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 a single integer n ($1 \le n \le 2 \cdot 10^5$).

The second line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($-10^9 \le b_i \le 10^9$).

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

Output

For each test case, print a single integer: the number of hybrid arrays $a_1, a_2, \ldots, a_n$ modulo $10^9 + 7$.

Example

input

```
4
3
1 -1 1
4
1 2 3 4
10
2 -1 1 -2 2 3 -5 0 2 -1
4
0 0 0 1
```

output

```
3
8
223
1
```

Note

In the first test case, the hybrid arrays are $[1, -2, 1]$, $[1, -2, 2]$, $[1, -1, 1]$.

In the second test case, the hybrid arrays are $[1, 1, 1, 1]$, $[1, 1, 1, 4]$, $[1, 1, 3, -1]$, $[1, 1, 3, 4]$, $[1, 2, 0, 1]$, $[1, 2, 0, 4]$, $[1, 2, 3, -2]$,

$[1, 2, 3, 4]$.

In the fourth test case, the only hybrid array is $[0, 0, 0, 1]$.