A. Multiples of Length

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

You are given an array a of n integers.

You want to make all elements of a equal to zero by doing the following operation **exactly three** times:

- Select a segment, for each number in this segment we can add a multiple of len to it, where len is the length of this segment (added integers can be different).

It can be proven that it is always possible to make all elements of a equal to zero.

Input

The first line contains one integer n ($1 \le n \le 100\,000$): the number of elements of the array.

The second line contains n elements of an array a separated by spaces: $a_1, a_2, \dots, a_n$ ($-10^9 \le a_i \le 10^9$).

Output

The output should contain six lines representing three operations.

For each operation, print two lines:

- The first line contains two integers l, r ($1 \le l \le r \le n$): the bounds of the selected segment.
- The second line contains $r-l+1$ integers $b_l, b_{l+1}, \dots, b_r$ ($-10^{18} \le b_i \le 10^{18}$): the numbers to add to $a_l, a_{l+1}, \dots, a_r$, respectively; b_i should be divisible by $r-l+1$.

Example

input

```
4
1 3 2 4
```

output

```
1 1
-1
3 4
4 2
2 4
-3 -6 -6
```

B. Stoned Game

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

T is playing a game with his friend, HL.

There are n piles of stones, the i-th pile initially has a_i stones.

T and HL will take alternating turns, with T going first. In each turn, a player chooses a non-empty pile and then removes a single stone from it. However, one cannot choose a pile that has been chosen in the previous turn (the pile that was chosen by the other player, or if the current turn is the first turn then the player can choose any non-empty pile). The player who cannot choose a pile in his turn loses, and the game ends.

Assuming both players play optimally, given the starting configuration of t games, determine the winner of each game.

Input

The first line of the input contains a single integer t $(1 \le t \le 100)$ — the number of games. The description of the games follows. Each description contains two lines:

The first line contains a single integer n $(1 \le n \le 100)$ — the number of piles.

The second line contains n integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 100)$.

Output

For each game, print on a single line the name of the winner, "T" or "HL" (without quotes)

Example

input
2 1 2 2 1 1

output
T HL

Note

In the first game, T removes a single stone from the only pile in his first turn. After that, although the pile still contains 1 stone, HL cannot choose from this pile because it has been chosen by T in the previous turn. Therefore, T is the winner.

C. Monster Invaders

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Ziota found a video game called "Monster Invaders".

Similar to every other shooting RPG game, "Monster Invaders" involves killing monsters and bosses with guns.

For the sake of simplicity, we only consider two different types of monsters and three different types of guns.

Namely, the two types of monsters are:

- a normal monster with 1 hp.
- a boss with 2 hp.

And the three types of guns are:

- Pistol, deals 1 hp in damage to one monster, r_1 reloading time
- Laser gun, deals 1 hp in damage to all the monsters in the current level (including the boss), r_2 reloading time
- AWP, instantly kills any monster, r_3 reloading time

The guns are initially not loaded, and the Ziota can only reload 1 gun at a time.

The levels of the game can be considered as an array $a_1, a_2, \ldots, a_n$, in which **the i-th stage has a_i normal monsters and 1 boss**. Due to the nature of the game, **Ziota cannot use the Pistol (the first type of gun) or AWP (the third type of gun) to shoot the boss before killing all of the a_i normal monsters**.

If Ziota damages the boss but does not kill it immediately, **he is forced to move out of the current level to an arbitrary adjacent level** (adjacent levels of level i $(1 < i < n)$ are levels $i-1$ and $i+1$, the only adjacent level of level 1 is level 2, the only adjacent level of level n is level $n-1$). Ziota can also choose to move to an adjacent level at any time. **Each move between adjacent levels are managed by portals with d teleportation time.**

In order not to disrupt the space-time continuum within the game, **it is strictly forbidden to reload or shoot monsters during teleportation.**

Ziota starts the game at level 1. The objective of the game is rather simple, to kill all the bosses in all the levels. He is curious about the minimum time to finish the game (assuming it takes no time to shoot the monsters with a loaded gun and Ziota has infinite ammo on all the three guns). Please help him find this value.

Input

The first line of the input contains five integers separated by single spaces: n $(2 \le n \le 10^6)$ — the number of stages, r_1, r_2, r_3 $(1 \le r_1 \le r_2 \le r_3 \le 10^9)$ — the reload time of the three guns respectively, d $(1 \le d \le 10^9)$ — the time of moving between adjacent levels.

The second line of the input contains n integers separated by single spaces $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^6, 1 \le i \le n)$.

Output

Print one integer, the minimum time to finish the game.

Examples

input
4 1 3 4 3 3 2 5 1

output

```
34
```

input

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

output

```
31
```

Note

In the first test case, the optimal strategy is:

- Use the pistol to kill three normal monsters and AWP to kill the boss (Total time $1 \cdot 3 + 4 = 7$)
- Move to stage two (Total time $7 + 3 = 10$)
- Use the pistol twice and AWP to kill the boss (Total time $10 + 1 \cdot 2 + 4 = 16$)
- Move to stage three (Total time $16 + 3 = 19$)
- Use the laser gun and forced to move to either stage four or two, here we move to stage four (Total time $19 + 3 + 3 = 25$)
- Use the pistol once, use AWP to kill the boss (Total time $25 + 1 \cdot 1 + 4 = 30$)
- Move back to stage three (Total time $30 + 3 = 33$)
- Kill the boss at stage three with the pistol (Total time $33 + 1 = 34$)

Note that here, we do not finish at level n, but when all the bosses are killed.

D. Rainbow Rectangles

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Shrimpy Duc is a fat and greedy boy who is always hungry. After a while of searching for food to satisfy his never-ending hunger, Shrimpy Duc finds M&M candies lying unguarded on a $L \times L$ grid. There are n M&M candies on the grid, the i-th M&M is currently located at $(x_i + 0.5, y_i + 0.5)$, and has color c_i out of a total of k colors (the size of M&Ms are insignificant).

Shrimpy Duc wants to steal a **rectangle** of M&Ms, specifically, he wants to select a rectangle with **integer** coordinates within the grid and steal all candies within the rectangle. Shrimpy Duc doesn't need to steal every single candy, however, he would like to steal **at least one candy for each color**.

In other words, he wants to select a rectangle whose sides are parallel to the coordinate axes and whose left-bottom vertex (X_1, Y_1) and right-top vertex (X_2, Y_2) are points with integer coordinates satisfying $0 \le X_1 < X_2 \le L$ and $0 \le Y_1 < Y_2 \le L$, so that for every color $1 \le c \le k$ there is at least one M&M with color c that lies within that rectangle.

How many such rectangles are there? This number may be large, so you only need to find it modulo $10^9 + 7$.

Input

The first line contains three positive integers n, k, L $(1 \le k \le n \le 2 \cdot 10^3, 1 \le L \le 10^9)$ — the number of M&Ms, the number of colors and the length of the grid respectively.

The following n points describe the M&Ms. Each line contains three integers x_i, y_i, c_i $(0 \le x_i, y_i < L, 1 \le c_i \le k)$ — the coordinates and color of the i-th M&M respectively.

Different M&Ms have different coordinates ($x_i \ne x_j$ or $y_i \ne y_j$ for every $i \ne j$), and for every $1 \le c \le k$ there is at least one M&M with color c.

Output

Output a single integer — the number of rectangles satisfying Shrimpy Duc's conditions, modulo $10^9 + 7$.

Examples

input

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

output

```
20
```

input

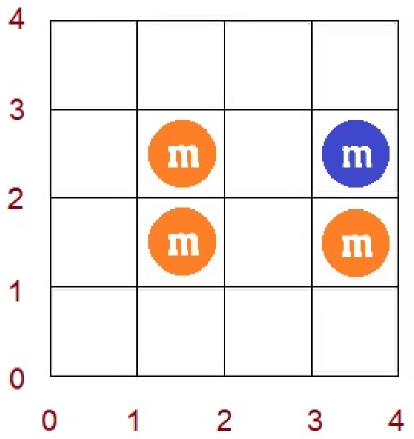
```
5 3 10
6 5 3
5 3 1
7 9 1
2 3 2
5 0 2
```

output
300

input
10 4 10 5 4 4 0 0 3 6 0 1 3 9 2 8 7 1 8 1 3 2 1 3 6 3 2 3 5 3 4 3 4
output
226

Note

Grid for the first sample:



E. Distance Matching

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given an integer k and a *tree* T with n nodes (n is even).

Let $dist(u, v)$ be the number of edges on the shortest path from node u to node v in T.

Let us define a *undirected weighted complete graph* $G = (V, E)$ as following:

- $V = \{x \mid 1 \le x \le n\}$ i.e. the set of integers from 1 to n
- $E = \{(u, v, w) \mid 1 \le u, v \le n, u \ne v, w = dist(u, v)\}$ i.e. there is an edge between every pair of distinct nodes, the weight being the distance between their respective nodes in T

Your task is simple, find a *perfect matching* in G with total edge weight k $(1 \le k \le n^2)$.

Input

The first line of input contains two integers n, k ($2 \le n \le 100\,000$, n is even, $1 \le k \le n^2$): number of nodes and the total edge weight of the perfect matching you need to find.

The i-th of the following $n-1$ lines contains two integers v_i, u_i ($1 \le v_i, u_i \le n$) denoting an edge between v_i and u_i in T. It is guaranteed that the given graph is a tree.

Output

If there are no matchings that satisfy the above condition, output "NO" (without quotes) on a single line.

Otherwise, you should output "YES" (without quotes) on the first line of output.

You should then output $\frac{n}{2}$ lines, the i-th line containing p_i, q_i ($1 \le p_i, q_i \le n$): the i-th pair of the matching.

Examples

input
4 2 1 2

```
2 3
3 4
```

output

```
YES
2 1
3 4
```

input

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

output

```
YES
3 1
2 4
```

Note

A *tree* is a connected acyclic undirected graph.

A *matching* is set of pairwise non-adjacent edges, none of which are loops; that is, no two edges share a common vertex.

A *perfect matching* is a matching which matches all vertices of the graph; that is, every vertex of the graph is incident to exactly one edge of the matching.