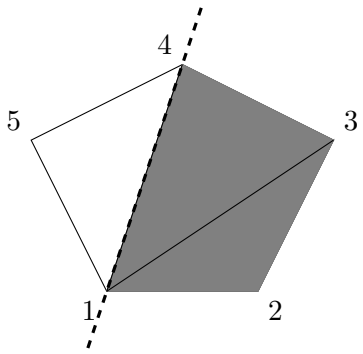
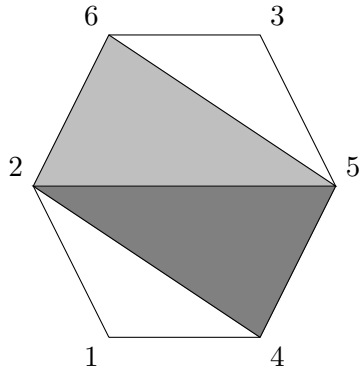
Triangulation

A *triangulation* of a polygon is a set of triangles with vertices at the vertices of a polygon. These triangles must not overlap and must cover the whole polygon.

We define a polygon *cut* as a straight line separating the polygon into two pieces.

Given a triangulated convex polygon, where each triangle has some color, find the maximal number of cuts one can do so that **no** two points of the same color end up in two different pieces.





Input

The input is read from standard input. The first line contains the number of vertices, n. Vertices are numbered with unique integers between 1 and n. Each of the next $n-2$ lines contains four integer numbers a, b, c and d ($1 \leq a, b, c, d \leq n$), meaning that the triangle which has its vertices in a, b and c has the color d. a, b and c are three different vertices. The input always contains data about a proper triangulation of a polygon and all triangles are colored.

Output

The program should write one line to standard output, containing one integer — the maximal number of cuts.

Example 1

Input	Output
5 1 2 3 2 4 5 1 1 3 1 4 2	1

Example 2

Input	Output
6 1 4 2 1 2 4 5 2 6 2 5 3 3 6 5 1	0

Constraints

$3 \leq n \leq 100{,}000$.

Grading

For test cases worth 50% of the total score, $n \leq 5000$