Distance Optimizing Triangulation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Consider a planar graph in the shape of a convex polygon with $2N$ vertices. Each vertex is numbered clockwise from 1 to $2N$. Currently, there are $2N$ bidirectional edges along the perimeter of the convex polygon. In other words, for every $1 \leq i \leq 2N$, vertex i and vertex $(i \pmod{2N}) + 1$ are connected by an edge.

Each vertex is colored with one of N colors numbered from 1 to N. For each color i ($1 \leq i \leq N$), there are exactly two vertices with color i. The indices of vertices with color i are x_i and y_i.

We want to add exactly $2N - 3$ bidirectional edges to this graph. After doing so, the following conditions must be satisfied:

- Each edge must connect two different vertices via a straight line segment.
- For any two distinct vertices, there can be at most one edge directly connecting the two vertices.
- The graph must still be planar.

Let $dist(a, b)$ be the length of the shortest path between two vertices a and b. Among all possible edge additions satisfying the above conditions, find one that minimizes $\sum_{i=1}^{N} dist(x_i, y_i)$.

Input

The first line contains a single integer N ($2 \leq N \leq 200\,000$).

Then, N lines are given. The i-th line contains two integers x_i, y_i denoting the indices of the two vertices with color i. ($1 \leq x_i, y_i \leq 2N$).

Output

On the first line, output the minimum possible value of $\sum_{i=1}^{N} dist(x_i, y_i)$.

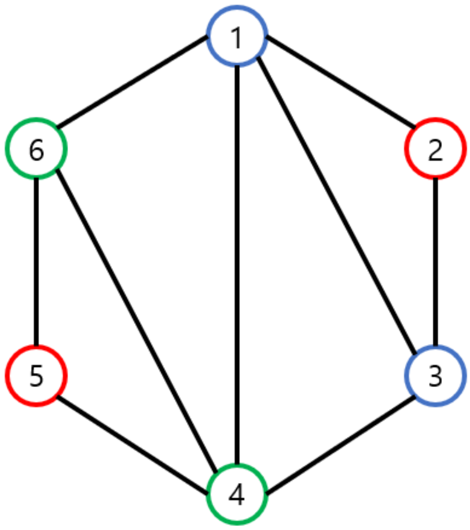
On the next $2N - 3$ lines, output two integers denoting the endpoints of each newly added edge.

If there are multiple answers, any of them will be accepted.

Example

standard input	standard output
3 1 3 2 5 6 4	5 1 3 4 1 4 6

Note


1
6
2
5
3
4