Laboratory work # 1

Student: HU Riqian
Student ID: 20321114
Timus Name: hduads2022_20321114
Mail: jhlxhrq@163.com

Problem # 1155

Screenshot from Timus:

9796435	10:17:46 29 Mar 2022	hduads2022_20321114	1155. Troubleduons	Java 1.8	Accepted		0.125	1 916 KB

Explanation of algorithm:

1. Find the points on the diagonal, depending on these, we can judge whether it is possible;

2. Fill the Array with the vertices;

3. Move all the vertices to A, E. Since if we can transfer all the vertices to A and E, we can annihilate them in couples.

Computational complexity of algorithm:

$$O(n);$$

Source code:

```
import java.io.PrintWriter;
import java.util.Scanner;

public class Troubleduons {

    // Define some constants in the question.
    static Scanner in = new Scanner(System.in);
    static PrintWriter out = new PrintWriter(System.out);
    static final String[] names = {"A", "B", "C", "D", "E", "F", "G", "H"};
    static int[] v = new int[8];

    public static void main(String[] args) {
        // Fill the data into the Array.
        for (int i = 0; i < v.length; i++) {
            v[i] = in.nextInt();
        }

        // Find whether the solution is possible.
        int opposite_1 = v[0] + v[2] + v[5] + v[7];
        int opposite_2 = v[1] + v[3] + v[4] + v[6];
```

```
        if (opposite_1 != opposite_2) {
            out.println("IMPOSSIBLE");
        } else {
            // Move all the vertices to A, E. Since if we can transfer all
the vertices
            // to A and E, we can annihilate them in couples.
            moveVertex(2, 0, 1);
            moveVertex(5, 0, 4);
            moveVertex(7, 0, 4);
            moveVertex(6, 4, 5);
            moveVertex(1, 4, 0);
            moveVertex(3, 4, 0);

            while (v[0]-- > 0) {
                out.println("AE-");
            }
        }
    }

    /**
     *
     * @param x : The specific vertex on the cube.
     * @param y : The specific vertex on the cube.
     * @param temp : Give the stopping point.
     */
    public static void moveVertex(int x, int y, int temp) {
        while(v[x] != 0) {
            if (v[temp] == 0) {
                v[temp]++;
                v[y]++;
                out.println(names[temp] + names[y] + "+");
            }
            v[temp]--;
            v[x]--;
            out.println(names[temp] + names[x] + "-");
        }
    }
}
```