Laboratory work # 2

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

Problem # 2025. Line Fighting

Screenshot from Timus:

9806241	20:20:49 3 Apr 2022	hduads2022_20321114	2025. Line Fighting	Java 1.8	Accepted		0.109	868 KB

Explanation of algorithm:

1. Divide the n fighters into k groups, which I divided them “averagely”, making the differences of each group are the smallest;

2. Match the fighters in the different groups, which I think it is can reach the maximal number of fights.

Computational complexity of algorithm:

$$O(n^2)$$

Source code:

```
import java.io.*;

public class LineFighting {
    static int NUMBER;
    static int[] FIGHTERS;
    static int[] GROUPS;

    public static void main(String[] args) throws IOException {
        new LineFighting().run();
    }

    StreamTokenizer in;
    PrintWriter out;

    int nextInt() throws IOException {
        in.nextToken();
        return (int)in.nval;
    }

    void run() throws IOException {
        in = new StreamTokenizer(new BufferedReader(new
InputStreamReader(System.in)));
```

```
        out = new PrintWriter(System.out);
        inputTest();
        outputResult(maximizeFights(NUMBER, divideGroup(NUMBER, FIGHTERS,
GROUPS)));
        out.flush();
    }

    void inputTest() throws IOException {
        NUMBER = nextInt();
        FIGHTERS = new int[NUMBER];
        GROUPS = new int[NUMBER];

        for (int i = 0; i < NUMBER; i++) {
            FIGHTERS[i] = nextInt();
            GROUPS[i] = nextInt();
        }
    }

    int[][] divideGroup(int number, int[] fighters, int[] groups) {
        int[][] dividedGroups = new int[number][];
        for (int i = 0; i < number; i++) {
            dividedGroups[i] = new int[groups[i]];
            int rounding = fighters[i] / groups[i];
            int reminder = fighters[i] - groups[i] * rounding;
            for (int j = 0; j < reminder; j++) {
                dividedGroups[i][j] = rounding + 1;
            }
            for (int j = reminder; j < groups[i]; j++) {
                dividedGroups[i][j] = rounding;
            }
        }
        return dividedGroups;
    }

    int[] maximizeFights(int number, int[][] group) {
        int[] maxFights = new int[number];
        for (int i = 0; i < number; i++) {
            int maxFight = 0;
            int reminder = FIGHTERS[i];
            for (int j = 0; j < GROUPS[i]; j++) {
                reminder -= group[i][j];
                maxFight += reminder * group[i][j];
            }
            maxFights[i] = maxFight;
        }
        return maxFights;
    }

    void outputResult(int[] array) {
        for (int i = 0; i < array.length; i++) {
            out.println(array[i]);
        }
    }
}
```