Exercise 5

Problem. **Intersecting Rectangles – Modification**

Scope

This exercise is aimed at discovering the principle “Changes are expected”.

As you can see, the problem from the exercise 4 is slightly revised here. Let’s have a look on how your implementation does change, and whether it is easy or not to modify your code according to the requirement modification. Pay particular attention to the underlined lines in this document: these lines correspond to the revised requirements.

Input data (revision)

Each input file line contains the following information describing a plane rectangle:

`#<rectangle-number> <rectangle-coordinates> <rectangle-color>`

<rectangle-number> and is a positive integer (should be unique for each rectangle).

<rectangle-coordinates> is a pair of integer coordinates representing two vertices at one of rectangle diagonals (assuming that rectangle’s sides are parallel to coordinate axes). Each pair is a pair of integer values (x-coordinate and y-coordinate separated by comma) in parentheses. One or more spaces are allowed as delimiters.

<rectangle-color> is a string representing color from the predefined set of allowed colors defined in the special configuration file “colors.txt”.

Data formats should correspond to the examples (see below).

Program (revision)

Your program finds the largest group of intersecting rectangles of the same color (i.e., 2 or more rectangles having at least one common point) and prints the answer formatted as shown in the example (including color information). The program calculates and prints the common area covered by all these rectangles (common area is the area covered by each from all the rectangles from the group) and the total area (total area is the area which is covered by at least one of the intersecting rectangles).

Notes (revision)

If there are no intersecting rectangles of the same color, the message “No intersection found” is printed.

Your program should check whether the color descriptions are properly used in the input file (i.e. only those colors are used which listed in the configuration file “colors.txt”). In a case of illegal color description found, the program prints an error message and stops. This message should include the invalid rectangle number.

Your program checks rectangle number uniqueness. If there are at least two rectangles with the same numbers, such an input is considered as illegal. In such a case, the program prints the information message and stops.

Implementation Requirements (revision)

In your solution, you should extend and/or modify the structural types you defined earlier, namely:

- A type representing a point in coordinate plane with the required fields for x-coordinate and y-coordinate;
- A type representing a rectangle in coordinate plane with the required fields for rectangle vertices, colors and additional fields that you consider useful.

Your program should be organized as a hierarchy of functions defined in several modules.

Example of input file:

```
#1 (1, 2) ( 4, 3 ) red
#4 (2, 5)  ( 5,  7 )   yellow
#2 (  -1,    4)  (4,6)  yellow
#7 (3, 0) (5, 2)  grey
#3  (2,2) (4,4) red
#25 (3,1) (4,9) yellow
```

Example of file “colors.txt”

```
white black red blue green orange yellow purple brown grey
```

Example of program output:

```
The largest group is a group of yellow rectangles
#2 #4 #25
Their common area is 1
Their total area is 19
```