Problem A: Manhattan Distance between Two Objects

(40% related to Lab 15)

Problem Description

Given two files, x.lef and b17.def where x.lef consists of some macro's and b17.def consists of some components, find the Manhattan distance between two components in b17.def and the sum of their perimeters. Here, each component is a reference to some macro in x.lef. The Manhattan distance between any two components B and C located respectively at (x_B, y_B) and (x_C, y_C) is defined as $|X_C - X_B| + |Y_C - Y_B|$. Each macro in x.lef starts with a keyword MACRO, followed by macro's name. Each macro has a size specified by its width and height. The perimeter of a macro is equal to twice the sum of its width and height. An example of macro's description can be found in x.lef.
A component is contained in the COMPONENTS section of b17.def and specified by a statement as follows:

- U1648 NOR2_xp33_75t + PLACED (674496 277920) FN ;

or

- FE_OFC71_FE_OFN37_reset BUF_x2_75t + SOURCE TIMING + PLACED (576000 576000) FS ;

or

- WELLTAP_1 TAPCELL_75t + SOURCE DIST + FIXED (92160 40320) FS

It starts with a -, followed by component's name for example U1648, and then followed by macro's name for example NOR2_xp33_75t. The two numbers in the parentheses are the coordinates of the component. The description of a component is ended with a ";". To process b17.def, you may use *getline*() function to read in a line at a time and convert the line into a *stringstream* in order to extract information from the line. To find the coordinates, you need scan the line until find the keyword "PLACED" or "FIXED" or the left parenthesis. Remember this is only a suggestion.

Input format

The input to the program will be several pairs of component's names. Each input line consists of a pair of component's names separated by a whitespace. Input is ended with a line containing only a zero.

Output format

The first line should present the total number of macros found in x.lef. The second line should give the total number of components in b17.def. It is then followed by the Manhattan distance between the two components in a pair and their perimeter, separated by a whitespace.

Requirements

You must use the following class definitions to store macro's and components, respectively.

```
class masterCell {
public:
    masterCell() { name = ""; perimeter = 0;}
    double getPerimeter() { return perimeter;}
    void setPerimeter(double p) {perimeter = p;}
    string getName() {return name;}
    void setName(string token) {name = token;}
```

```
private:
    string name;    // Name of a macro
    double perimeter;    // perimeter of a macro
};

class Component {
public:
    Component() { name = ""; x = 0; y =0; perimeter = 0;}
    double getPerimeter() { return perimeter;}
    void setPerimeter(double p) {perimeter = p;}
    string getName() {return name;}
    void setName(string token) {name = token;}
    void setX(int xc) {x = xc;}
    void setY(int yc) {y = yc;}
    int getX() { return x;}
    int getY() { return y;}
private:
    string name;
    int x;    // x coordinate
    int y; // y coordinate
    double perimeter; // perimeter of a component
};
```

Example Input:

U1648 FE_OFC71_FE_OFN37_reset
WELLTAP_1 U1648
FE_OFC71_FE_OFN37_reset WELLTAP_1
0

Example Output:

217
27327
396576 8.208
819936 6.912
1019520 7.344