**算法分析与设计实验报告**

**第 2 次实验**

**10081 Crossed ladders**

|  |
| --- |
| **Problem description** |
| A narrow street is lined with tall buildings. An x foot long ladder is rested at the base of the building on the right side of the street and leans on the building on the left side. A y foot long ladder is rested at the base of the building on the left side of the street and leans on the building on the right side. The point where the two ladders cross is exactly c feet from the ground. How wide is the street? |
| **Input** |
| Each line of input contains three positive floating point numbers giving the values of x, y, and c. |
| **Output** |
| For each line of input, output one line with a floating point number giving the width of the street in feet, with three decimal digits in the fraction. |
| **Sample Input** |
| 30 40 10  12.619429 8.163332 3  10 10 3  10 10 1 |
| **Sample Output** |
| 26.033  7.000  8.000  9.798 |

**附录：完整代码**

*//Crossed ladders*

#include <iostream>

#include <cstdio>

#include <cmath>

using namespace std;

double x, y, c, mid, l, r;

double f(double m)

{

    return (1 / sqrt(x \* x - m \* m) + 1 / sqrt(y \* y - m \* m) - 1 / c);

}

int main()

{

    while (scanf("%lf%lf%lf", &x, &y, &c) == 3)

    {

        mid = 0;

        l = 0;

        r = min(x, y);

        while (r - l >= 1e-9)

        {

            mid = (l + r) / 2.0;

            if (f(mid) > 0)

            {

                r = mid;

            }

            if (f(mid) < 0)

            {

                l = mid;

            }

        }

        printf("%0.3f\n", mid);

    }

    return 0;

}

*//Crossed ladders测试数据生成器*

#include <iostream>

#include <fstream>

#include <time.h>

#define random(a, b) (rand() % (b - a + 1) + a)

using namespace std;

int main()

{

    int t;

    cin >> t;

    ofstream outfile("Crossed ladders\_in.txt", ios::out);

    srand((int)time(NULL));

    for (int i = 1; i <= t; i++)

    {

        outfile << random(25, 40) << " ";

        outfile << random(25, 40) << " ";

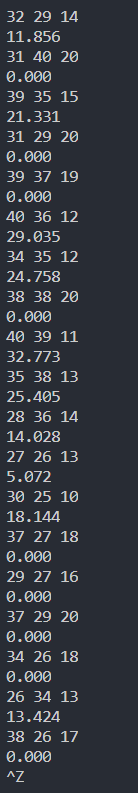
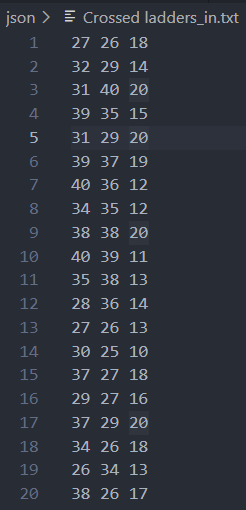
        outfile << random(10, 20) <<endl;

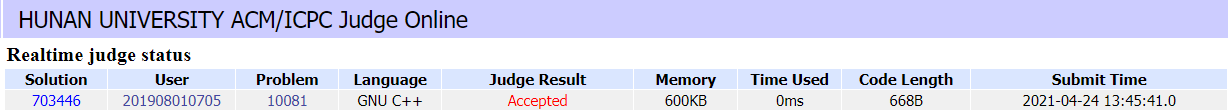
    }

    outfile.close();

    return 0;

}



****

****