

Java-数组循环-给定一组不等式

题目描述:

给定一组不等式, 判断是否成立并输出不等式的最大差(输出浮点数的整数部分), 要求: 1) 不等式系数为 double 类型, 是一个二维数组; 2) 不等式的变量为 int 类型, 是一维数组; 3) 不等式的目标值为 double 类型, 是一维数组; 4) 不等式约束为字符串数组, 只能是: ">",">=","<","<=","=", 例如, 不等式组:

$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 \leq b_1;$

$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + a_{25}x_5 \leq b_2;$

$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + a_{35}x_5 \leq b_3;$

最大差 $= \max\{ (a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 - b_1), (a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + a_{25}x_5 - b_2), (a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + a_{35}x_5 - b_3) \}$, 类型为整数(输出浮点数的整数部分)

输入描述:

1) 不等式组系数(double 类型):

$a_{11}, a_{12}, a_{13}, a_{14}, a_{15}$

$a_{21}, a_{22}, a_{23}, a_{24}, a_{25}$

$a_{31}, a_{32}, a_{33}, a_{34}, a_{35}$

2) 不等式变量(int 类型):

x_1, x_2, x_3, x_4, x_5

3) 不等式目标值(double 类型): b_1, b_2, b_3

4) 不等式约束(字符串类型): $\leq, \geq, =$

输入:
 $a_{11}, a_{12}, a_{13}, a_{14}, a_{15}; a_{21}, a_{22}, a_{23}, a_{24}, a_{25}; a_{31}, a_{32}, a_{33}, a_{34}, a_{35}; x_1, x_2, x_3, x_4, x_5; b_1, b_2, b_3; \leq, \geq, =$

输出描述:

true 或者 false, 最大差

补充说明:

示例 1

输入:

2.3,3,5.6,7,6;11,3,8.6,25,1;0.3,9,5.3,66,7.8;1,3,2,7,5;340,670,80.6;<=<=<=

输出:

false 458

说明:

示例 2

输入:

2.36,3,6,7.1,6;1,30,8.6,2.5,21;0.3,69,5.3,6.6,7.8;1,13,2,17,5;340,67,300.6;<=>=<=

输出:

false 758

说明:

import java.util.Scanner;

// 注意类名必须为 Main, 不要有任何 package xxx 信息

```

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String[] questions=in.nextLine().split(";");
        double[][] d1=new double[3][5];
        for(int i=0;i<3;i++){
            String[] tmp=questions[i].split(",");
            for(int j=0;j<tmp.length;j++){
                d1[i][j]=Double.valueOf(tmp[j]);
            }
        }
        int[] d2=new int[5];
        String[] s2=questions[3].split(",");
        for(int i=0;i<5;i++){
            d2[i]=Integer.valueOf(s2[i]);
        }
        double[] d3=new double[3];
        String[] s3=questions[4].split(",");
        for(int i=0;i<3;i++){
            d3[i]=Double.valueOf(s3[i]);
        }
        String[] d4=questions[5].split(",");
        double[] a1=new double[3];
        for(int i=0;i<3;i++){
            a1[i]=0;
            for(int j=0;j<5;j++){
                a1[i]+=d1[i][j]*d2[j];
            }
        }
        int ans=Integer.MIN_VALUE;
        boolean p=true;
        for(int i=0;i<3;i++){
            //System.out.println(a1[i]+" "+d3[i]+" "+d4[i]);
            double tmp=a1[i]-d3[i];
            if(tmp==0){
                ans=Math.max(ans,0);
                if(p){
                    if(!d4[i].equals("=")&&!d4[i].equals(">=")&&!d4[i].equals("<=")) p=false;
                }
            }
            else if(tmp>0){
                ans=Math.max(ans,(int)Math.floor(tmp));
                if(p){
                    if(!d4[i].equals(">")&&!d4[i].equals(">=")) p=false;
                }
            }
        }
    }
}

```

```
        }
    }
    else{
        tmp=-Math.floor(-tmp);
        ans=Math.max(ans,(int)tmp);
        if(p){
            if(!d4[i].equals("<")&&!d4[i].equals("<=")) p=false;
        }
    }
}
if(p) System.out.print("true "+ans);
else System.out.print("false "+ans);
}
}
```