

标题：承压计算 [详解] X星球的高科技实验室中整齐地堆放着某批珍贵金属原料。 每块金属原料的外形、尺寸完全一致，但重量不同。 金属材料被严格地堆放成金字塔形。

原创

更多文章：[博客->百度搜索：渣渣龙网站](#) 于 2019-11-24 21:25:48 发布 1929 收藏 1
分类专栏：[蓝桥杯](#) 文章标签：[蓝桥杯](#)
版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/qq_43044661/article/details/103229211

版权



[蓝桥杯 专栏收录该内容](#)

37 篇文章 6 订阅
订阅专栏
标题：承压计算

X星球的高科技实验室中整齐地堆放着某批珍贵金属原料。
每块金属原料的外形、尺寸完全一致，但重量不同。
金属材料被严格地堆放成金字塔形。

```

      7
     5 8
    7 8 8
   9 2 7 2
  8 1 4 9 1
 8 1 8 8 4 1
7 9 6 1 4 5 4
5 6 5 5 6 9 5 6
5 5 4 7 9 3 5 5 1
7 5 7 9 7 4 7 3 3 1
4 6 4 5 5 8 8 3 2 4 3
1 1 3 3 1 6 6 5 5 4 4 2
9 9 9 2 1 9 1 9 2 9 5 7 9
4 3 3 7 7 9 3 6 1 3 8 8 3 7
3 6 8 1 5 3 9 5 8 3 8 1 8 3 3
8 3 2 3 3 5 5 8 5 4 2 8 6 7 6 9
8 1 8 1 8 4 6 2 2 1 7 9 4 2 3 3 4
2 8 4 2 2 9 9 2 8 3 4 9 6 3 9 4 6 9
7 9 7 4 9 7 6 6 2 8 9 4 1 8 1 7 2 1 6
9 2 8 6 4 2 7 9 5 4 1 2 5 1 7 3 9 8 3 3
5 2 1 6 7 9 3 2 8 9 5 5 6 6 6 2 1 8 7 9 9
6 7 1 8 8 7 5 3 6 5 4 7 3 4 6 7 8 1 3 2 7 4
2 2 6 3 5 3 4 9 2 4 5 7 6 6 3 2 7 2 4 8 5 5 4
7 4 4 5 8 3 3 8 1 8 6 3 2 1 6 2 6 4 6 3 8 2 9 6
1 2 4 1 3 3 5 3 4 9 6 3 8 6 5 9 1 5 3 2 6 8 8 5 3
2 2 7 9 3 3 2 8 6 9 8 4 4 9 5 8 2 6 3 4 8 4 9 3 8 8
```

777975279251926539357354289
7766875582477472692182985736
59455755635395895412614353241
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

其中的数字代表金属块的重量（计量单位较大）。
最下一层的X代表30台极高精度的电子秤。

假设每块原料的重量都十分精确地平均落在下方的两个金属块上，
最后，所有的金属块的重量都严格精确地平分落在最底层的电子秤上。
电子秤的计量单位很小，所以显示的数字很大。

工作人员发现，其中读数最小的电子秤的示数为：2086458231

请你推算出：读数最大的电子秤的示数为多少？

注意：需要提交的是一个整数，不要填写任何多余的内容。

笨笨有话说：

不断的除2，加到下面，除2,加到下面,... 不会浮点精度溢出吧？

歪歪有话说：

怕除不开还不好办， 把每个数字扩大一定的倍数不就好了。
位

```
package eight;

import java.util.Scanner;
public class cyjs {
public static void main(String[] args) {
    Scanner shu=new Scanner(System.in);
    double [][]dou=new double[30][30];
    for (int i = 0; i < 29; i++) {
        for (int j = 0; j < i+1 && j < 29; j++) { //j<i+1 每行固定读取几个数
            dou[i][j] = shu.nextDouble();
        }
    }
    for (int i = 0; i <=29; i++) {
        if(i==29) break; //最后一行不进行下一步的累加 因为下一步没有值
        for (int j = 0; j < i+1 && j <=29; j++) {
            dou[i+1][j]=dou[i+1][j]+dou[i][j]/2.0; //下一行的左支撑
            dou[i+1][j+1]=dou[i+1][j+1]+dou[i][j]/2.0; //下一行的右支撑
        }
    }
    double min=1000;
    double max=0;
    for(int i=0;i<=29;i++) {
        min=Math.min(min, dou[29][i]);
        max=Math.max(max, dou[29][i]);
    }
    System.out.println(min+" "+max);
    System.out.println((2086458231/min)*max);
    //通过给出的最小值算出该星球的换算单位
    //结果:72665192664
}
```