

拼图

jigsaw

【问题描述】

5岁的小P对剪纸很感兴趣，他总是喜欢把一个矩形的纸片剪成一个又一个的凸多边形。但是，每一次剪完后，他总是怀疑自己弄丢了一些纸片。聪明的他想到了一个方法来检测纸片是否弄丢：他将这些凸多边形拼起来，如果能够拼成一个矩形，他就认为纸片没有弄丢。由于纸片的数量不是很多，这个工作并不难。但是，久而久之，他对这项工作不感兴趣了，所以，他找到了你，希望你能够告诉他，这些凸多边形纸片能不能够拼成矩形。

【输入文件】

第一行只有一个正整数 n ($1 \leq n \leq 8$)，表示凸多边形的个数。

以下 n 行每一行描述一个凸多边形，格式如下：

第 $i+1$ 行的第一个数 m_i ($3 \leq m_i \leq 8$) 表示凸多边形的点数，接下来有 m_i 对实数，一对实数给出了一个点的坐标，这 m_i 个顶点按照从任意一个顶点出发的逆时针顺序给出。且所有实数都在 $(-1000, 1000)$ 的范围内，小数点后不超过 8 位。

【输出文件】

如果不能拼成矩形，输出只有一行 “No”。

如果能拼成矩形，输出的第一行为 “Yes”。

接下来的 n 行描述拼法。

如果能够拼成一个 $X*Y$ 的矩形，那么矩形的四个顶点的坐标是 $(0,0)$ 、 $(0,Y)$ 、 (X,Y) 、 $(X,0)$ 。这 n 行输出每一个凸多边形的顶点的坐标（拼成矩形后）。按照输入的顺序，即第一个输出的凸多边形对应输入的第一个凸多边形。对于每一个凸多边形，输出也按照输入的顺序，即一个多边形的第一个顶点对应输入的第一个顶点。这样，输出总共有 n 行，第 i 行有 m_i 对数。

【输入样例】

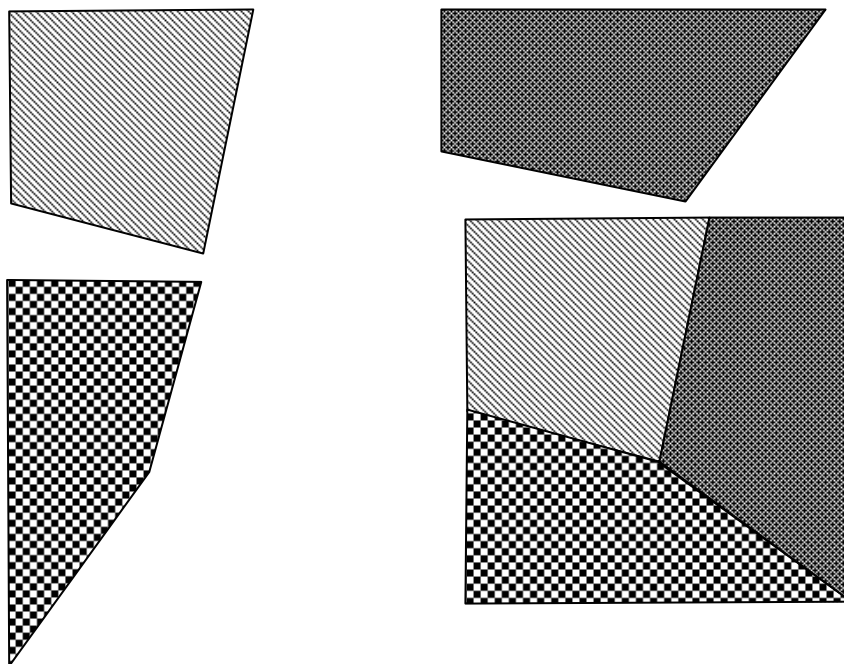
```
3
4 0 0 4 -1 5 4 0 4
4 0 0 5 -1 8 3 0 3
4 0 0 0 -8 3 -4 4 0
```

【输出样例】

```
Yes
0 4 4 3 5 8 0 8
5 8 4 3 8 0 8 8
0 0 8 0 4 3 0 4
```

【样例说明】

如下图，左上、右上和左下描述了输入的凸多边形，右下描述了输出的矩形。



【注意】

由于矩形纸片的两面的颜色不同，所以纸片只能旋转和平移，不能翻转。所以，输出的 m_i 个顶点也应该是逆时针顺序的。

【评分说明】

两个实数之差的绝对值小于 10^{-7} 时认为两个实数相等。

【你如何测试自己的程序】

- 在工作目录下建立输入文件 `jigsaw.in`，写入你自己设计的输入数据。
- 执行你的程序，产生输出文件 `jigsaw.out`。
- 执行检查程序

```
./jigsaw_c
```

该程序检查 `jigsaw.out` 中的方案是否合法，并把检查结果打印到屏幕。