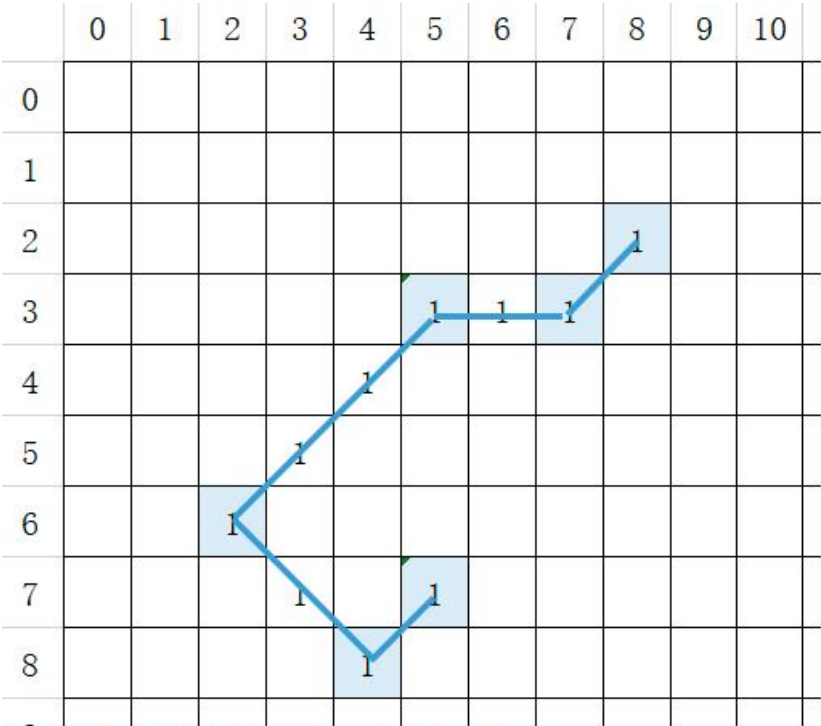


题目描述：

下图中，每个方块代表一个像素，每个像素用其行号和列号表示。



为简化处理，多段线的走向只能是水平、竖直、斜向45度。
上图中的多段线可以用下面的坐标串表示：(2, 8), (3, 7), (3, 6), (3, 5), (4, 4), (5, 3), (6, 2), (7, 3), (8, 4), (7, 5)。

但可以发现，这种表示不是最简的，其实只需要存储6个蓝色的关键点即可，它们是线段的起点、拐点、终点，而剩下4个点是冗余的。

现在，请根据输入的包含有冗余数据的多段线坐标列表，输出其最简化的结果。

输入描述：

2 8 3 7 3 6 3 5 4 4 5 3 6 2 7 3 8 4 7 5

- 1、所有数字以空格分隔，每两个数字一组，第一个数字是行号，第二个数字是列号；
- 2、行号和列号范围为[0, 64)，用例输入保证不会越界，考生不必检查；
- 3、输入数据至少包含两个坐标点。

输出描述：

2 8 3 7 3 5 6 2 8 4 7 5

压缩后的最简化坐标列表，和输入数据的格式相同。

补充说明：

输出的坐标相对顺序不能变化。

示例 1

输入：

2 8 3 7 3 6 3 5 4 4 5 3 6 2 7 3 8 4 7 5

输出：

2 8 3 7 3 5 6 2 8 4 7 5

说明：

如上图所示，6个蓝色像素的坐标依次是（2,8）、（3,7）、（3,5）、（6,2）、（8,4）、（7,5）。

将他们按顺序出即可。

```
import sys
```

```
result = []
```

```
for line in sys.stdin:
```

```
    a = line.replace('\n','').split(' ')
```

```
    temp = [int(a[0]),int(a[1])]
```

```
    temp_dif = [0,0]
```

```
    result.append(a[0])
```

```
    result.append(a[1])
```

```
    for i in range(2,len(a),2):
```

```
        x = int(a[i])
```

```
        y = int(a[i+1])
```

```
        x_diff = x - temp[0]
```

```
        y_diff = y-temp[1]
```

```
        temp =[x,y]
```

```
        if not(temp_dif[0] == 0 and temp_dif[1] == 0) and x_diff * temp_dif[1] == y_diff *
```

```
temp_dif[0]:
```

```
            result.pop()
```

```
            result.pop()
```

```
            temp_dif = [x_diff,y_diff]
```

```
            result.append(a[i])
```

```
            result.append(a[i+1])
```

```
print(' '.join(result))
```