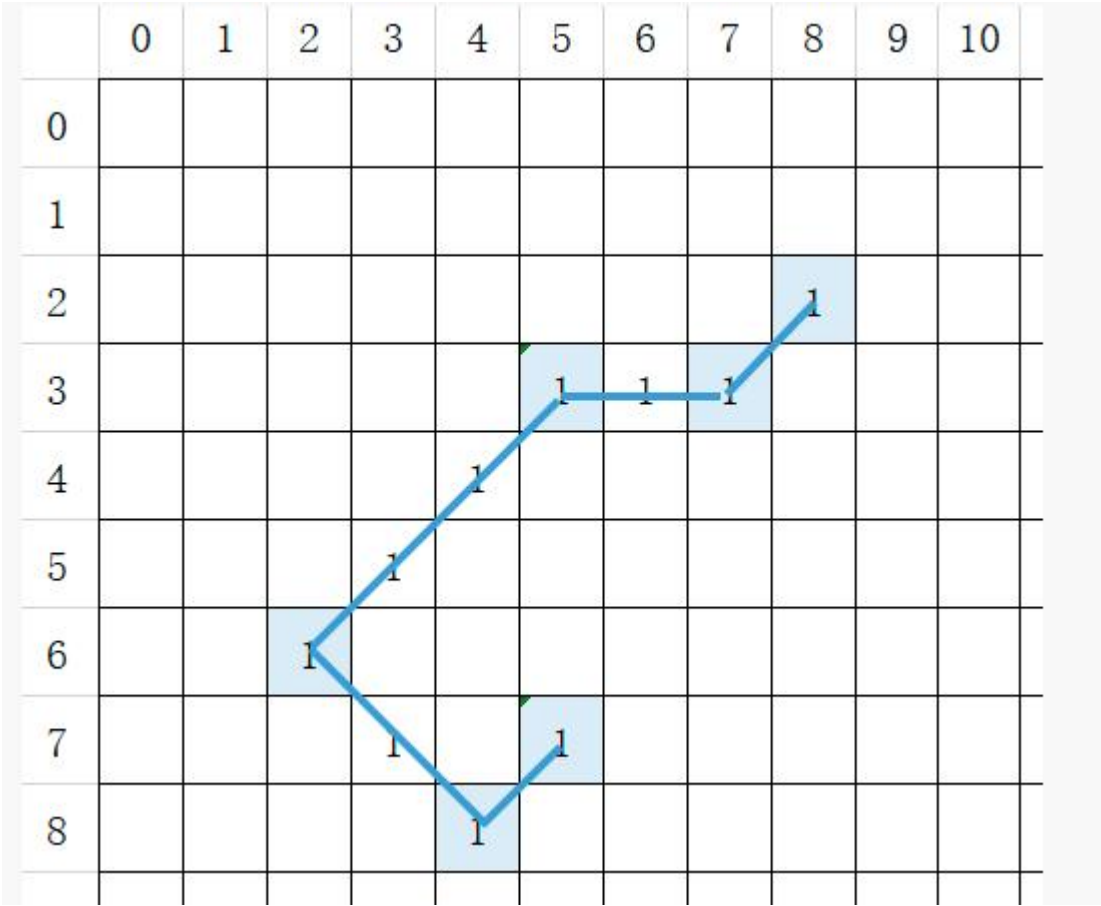


题目描述：

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。

```
#include <stdio.h>
```

```
int main() {
    int x[10000];
    int y[10000];
    int flag[10000];

    int xx=0;
    int yy=0;
    int i=0;
    while(scanf("%d %d",&x[i],&y[i])!=EOF){
        if(i==0){
            flag[i]=1;
        }
        else if(i==1){
            xx=x[i]-x[i-1];
            yy=y[i]-y[i-1];
        }
        else{
```

```
        int xxx=x[i]-x[i-1];
        int yyy=y[i]-y[i-1];
        if(xxx!=xx || yyy!=yy){
            flag[i-1]=1;
            xx=xxx;
            yy=yyy;
        }
    }
    i++;

}
flag[i-1]=1;

for(int j=0;j<i;j++){
    if(flag[j]==1) printf("%d %d ",x[j],y[j]);
}

return 0;
}
```