

题目描述：

AI 识别到面板上有 N ($1 \leq N \leq 100$) 个指示灯，灯大小一样，任意两个灯之间无重叠。由于 AI 识别误差，每次识别到的指示灯位置可能有差异，以 4 个坐标值描述 AI 识别的指示灯的大小和位置(左上角 $x1,y1$ ，右下角 $x2,y2$)，

请输出先行后列排序的指示灯的编号，排序规则：

- 1、每次在尚未排序的灯中挑选最高的灯作为的基准灯，
- 2、找出和基准灯属于同一行所有的灯进行排序。两个灯高低偏差不超过灯半径算同一行（即两个灯 y 坐标的差 \leq 灯高度的一半）。

输入描述：

第一行为 N ，表示灯的个数

接下来 N 行，每行为 1 个灯的坐标信息，格式为：编号 $x1\ y1\ x2\ y2$ ，编号全局唯一， $1 \leq$ 编号 ≤ 100 ， $0 \leq x1 < x2 \leq 1000$ ， $0 \leq y1 < y2 \leq 1000$

输出描述：

排序后的编号列表，编号之间以空格分隔

补充说明：

示例

示例 1

输入：

5

1 0 0 2 2

2 6 1 8 3

3 3 2 5 4

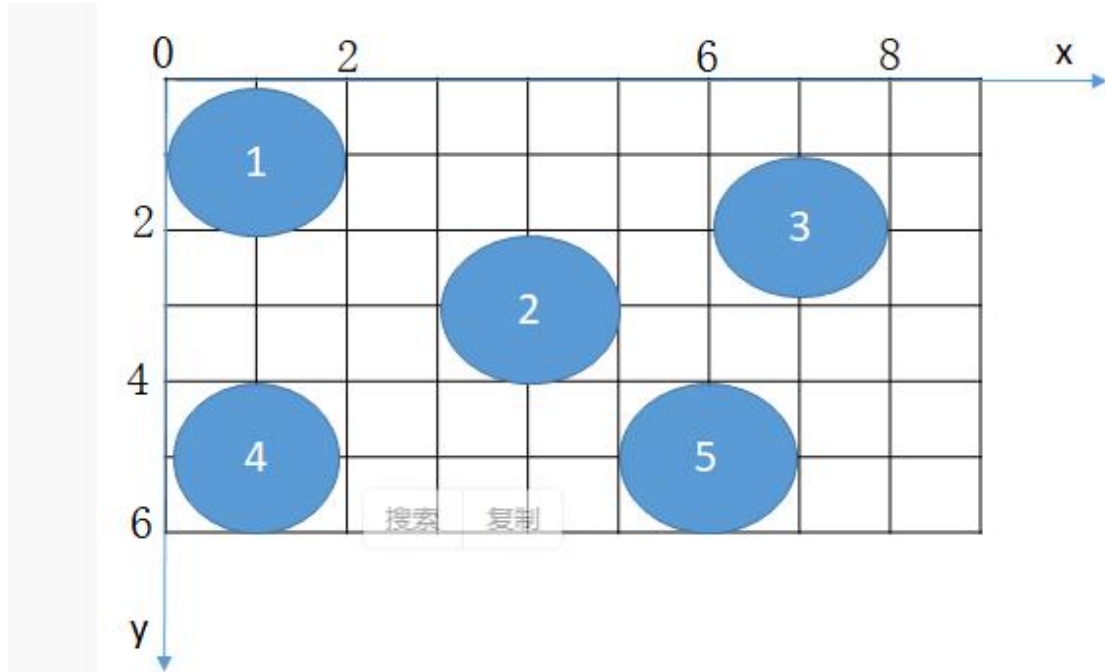
5 5 4 7 6

4 0 4 2 6

输出：

1 2 3 4 5

说明：



```
import java.util.*;
```

```
public class Main{
```

```
    public static void main(String[] args){
        Scanner in=new Scanner(System.in);
        int N=in.nextInt();
        ArrayList<led> larr=new ArrayList<led>();
        for(int i=0;i<N;i++){
            larr.add(new led());
            larr.get(i).num=in.nextInt();
            for(int j=0;j<4;j++){
                larr.get(i).xy[j]=in.nextInt();
            }
        }
        int rad=(larr.get(0).xy[3]-larr.get(0).xy[1])/2;
        larr.sort((a,b)->(a.xy[0]-b.xy[0]));
        larr.sort((a,b)->(a.xy[1]-b.xy[1]));
        ArrayList<led> resarr=new ArrayList<led>();
        int cursize=0;
        while(!larr.isEmpty()) {
            int x=larr.get(0).xy[1];
            resarr.add(larr.get(0));
            larr.remove(0);
            while(!larr.isEmpty()&&larr.get(0).xy[1]-x<=rad) {
```

```

        boolean flag=false;
        for(int j=cursize;j<resarr.size();j++) {
            if(larr.get(0).xy[0]<resarr.get(j).xy[0]) {
                resarr.add(j,larr.get(0));
                flag=true;
                break;
            }
        }
        if(!flag) {
            resarr.add(larr.get(0));
        }
        larr.remove(0);
    }
    cursize=resarr.size();
}
for(int i=0;i<N;i++) {
    System.out.print(resarr.get(i).num+" ");
}
}
static class led{
    public int[] xy=new int[4];
    public int num=0;
}
}

```