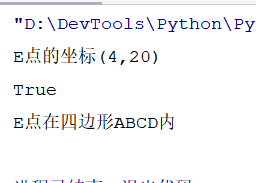
# 第一题：

效果：

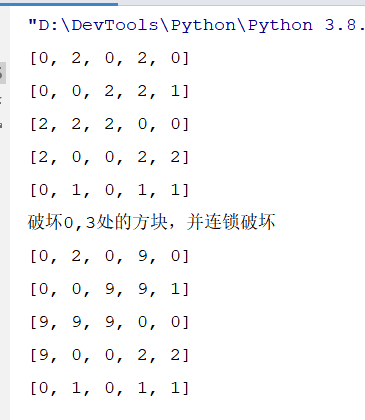


python代码：

**import** random  
**def** relationship(x, y, verts):  
 **try**:  
 x, y = float(x), float(y)  
 **except**:  
 **return False** *# 获取横坐标和纵坐标的值* vertx = [xyvert[0] **for** xyvert **in** verts]  
 verty = [xyvert[1] **for** xyvert **in** verts]  
 *# ABCD四个点中，通过其横坐标和纵坐标的最大值和最小值，初步判断目标坐标点是否有可能在这个四边形之内* **if not** verts **or not** min(vertx) <= x <= max(vertx) **or not** min(verty) <= y <= max(verty):  
 **return False** *# 进一步判断E的与ABCD点的坐标关系* nvert = len(verts) - 1  
 is\_in = **False** j = nvert  
 **for** i **in** range(nvert):  
 **if** i != 0:  
 j = i  
 i = i + 1  
 **if** ((verty[i] > y) != (verty[j] > y)) **and** (  
 x < (vertx[j] - vertx[i]) \* (y - verty[i]) / (verty[j] - verty[i]) + vertx[i]):  
 is\_in = **not** is\_in  
  
 print(is\_in)  
 **return** is\_in  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 *# 假设ABCD点的二维坐标为* quadrangle = [(40, 0), (0, 0), (0, 40), (40, 40)]  
 *# E点的坐标(x,y)* x = random.randint(0, 50)  
 y = random.randint(0, 50)  
 print(**"E点的坐标(%d,%d)"** % (x, y))  
 **if** relationship(x, y, quadrangle):  
 print(**"E点在四边形ABCD内"**)  
 **else**:  
 print(**"E点不在四边形ABCD内"**)

# 第二题：

效果



python代码：

size = 5  
*# 连锁挖矿***def** block(x, y, block\_list):  
 k = block\_list[x][y];  
 recursion(x, y, k, block\_list)  
*# 遍历***def** recursion(x, y, k, block\_list):  
 **if** (x < 0 **or** x >= len(block\_list)):  
 **return  
 if** (y < 0 **or** y >= len(block\_list)):  
 **return  
 if** (block\_list[x][y] != k):  
 **return  
 if** (block\_list[x][y] == 9):  
 **return** block\_list[x][y] = 9;  
 recursion(x + 1, y, k, block\_list)  
 recursion(x - 1, y, k, block\_list)  
 recursion(x, y + 1, k, block\_list)  
 recursion(x, y - 1, k, block\_list)  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 block\_list = [[0, 2, 0, 2, 0], [0, 0, 2, 2, 1], [2, 2, 2, 0, 0], [2, 0, 0, 2, 2], [0, 1, 0, 1, 1]]  
 **for** i **in** block\_list:  
 print(i)  
 x = 0  
 y = 3  
 print(**"破坏"** + str(x) + **","** + str(y) + **"处的方块，并连锁破坏"**)  
 *# block方法* block(x, y, block\_list)  
 **for** i **in** block\_list:  
 print(i)