

题目描述：

周末小明准备去爬山锻炼， 0 代表平地，山的高度使用 1 到 9 来表示，小明每次爬山或下山高度只能相差 k 及 k 以内，每次只能上下左右一个方向上移动一格，小明从左上角 $(0,0)$ 位置出发

输入描述：

第一行输入 $m\ n\ k$ （空格分隔），代表 $m*n$ 的二维山地图， k 为小明每次爬山或下山高度差的最大值。然后接下来输入山地图，一共 m 行 n 列，均以空格分隔。

取值范围： $0 < m \leq 500, 0 < n \leq 500, 0 < k \leq 5$

输出描述：

请问小明能爬到的最高峰多高，到该最高峰的最短步数，输出以空格分隔。同高度的山峰输出较短步数。如果没有可以爬的山峰则高度和步数都返回 0 。

补充说明：

所有用例输入均为正确格式，且在取值范围内，考生不需要考虑不合法的输入格式。

示例 1

输入：

```
5 4 1
0 1 2 0
1 0 0 0
1 0 1 2
1 3 1 0
0 0 0 9
```

输出：

```
2 2
```

说明：

输出解读：根据山地图可知，能爬到的最高峰在(0,2)位置，高度为 2，最短路径为 (0,0)-(0,1)-(0,2)，最短步数为 2。

示例 2

输入：

```
5 4 3
0 0 0 0
0 0 0 0
0 9 0 0
0 0 0 0
0 0 0 9
```

输出：

```
0 0
```

说明：

输出解读：根据山地图可知，每次爬山距离 3，无法爬到山峰上，步数为 0。

```
m, n, k = [int(x) for x in input().strip().split(' ')]
grid = []
for i in range(m):
    grid.append([int(x) for x in input().strip().split(' ')])

step = [[-1] * n for _ in range(m)]
step[0][0] = 0
head, tail = 0, 1
q = [[0,0]]

dx=[0, 1, 0, -1]
dy=[1, 0, -1, 0]
while head < tail:
    for nowx, nowy in q[head : tail]:
        for index in range(4):
            x = nowx + dx[index]
            y = nowy + dy[index]
            if x >= 0 and x < m and y >= 0 and y < n and abs(grid[x][y] - grid[nowx][nowy])
<= k and step[x][y] == -1:
                step[x][y] = step[nowx][nowy] + 1
                q.append([x, y])

    head, tail = tail, len(q)
```

```
res=[0, 0]
for i in range(m):
    for j in range(n):
        if step[i][j] >= 0:
            if res[0] < grid[i][j] or (res[0] == grid[i][j] and res[1] > step[i][j]):
                res = [grid[i][j], step[i][j]]

print(res[0], res[1])
```