

C-周末爬山-周末小明准备去爬山锻炼

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。

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

```
int max = 0;
```

```
int m, n, k;
```

```
int step = 0;
```

```
int max_height = 0;
```

```
int arr[500][500];
```

```
void calc(int a, int b, int last, int depth);
```

```
void dfs(int a, int b, int last, int depth) {
```

```
    if (a < 0 || b < 0 || a >= m || b >= n ||
```

```
        arr[a][b] == -1 || last + k < arr[a][b] || max_height == max && depth > step) {
```

```
        return;
```

```
    }
```

```
    depth++;
```

```

        if (max_height < arr[a][b]) {
            max_height = arr[a][b];
            step = depth;
        }
        if (max_height == arr[a][b] && depth < step) {
            step = depth;
        }

        calc(a, b, arr[a][b], depth);
    }

void calc(int a, int b, int last, int depth) {
    int t = arr[a][b];
    arr[a][b] = -1;
    dfs(a, b + 1, last, depth);
    dfs(a, b - 1, last, depth);
    dfs(a + 1, b, last, depth);
    dfs(a - 1, b, last, depth);
    arr[a][b] = t;
}

int main() {
    scanf("%d%d%d", &m, &n, &k);
    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < n; ++j) {
            scanf("%d", &arr[i][j]);
            max = max > arr[i][j] ? max : arr[i][j];
        }
    }
    calc(0, 0, arr[0][0], 0);
    printf("%d %d\n", max_height, step);
    return 0;
}

```