

题目描述：

有一个大小是 $N \times M$ 的战场地图，被墙壁 '#' 分隔成大小不同的区域，上下左右四个方向相邻的空地 '.' 属于同一个区域，只有空地上可能存在敌人 'E'，请求出地图上总共有多少区域里的敌人数小于 K 。

输入描述：

第一行输入为 N, M, K ;

N 表示地图的行数， M 表示地图的列数， K 表示目标敌人数量 $N, M \leq 100$;

之后为一个 $N \times M$ 大小的字符数组。

输出描述：

敌人数小于 K 的区域数量

补充说明：

示例1

输入：3 5 2

..#EE

E.#E.

###..

输出：1

说明：地图被墙壁分为两个区域，左边区域有1个敌人，右边区域有3个敌人，符合条件的区域数量是1

```

1 import java.util.*;
2
3 // 注意类名必须为 Main, 不要有任何 package xxx 信息
4 public class Main {
5     static int n,m,k;
6     static int[][] dirs = {{-1,0},{1,0},{0,-1},{0,1}};
7     static boolean[][] visited;
8     static char[][] arr;
9     public static void main(String[] args) {
10         Scanner cin = new Scanner(System.in);
11         String[] s = cin.nextLine().split(" ");
12         n = Integer.parseInt(s[0]);
13         m = Integer.parseInt(s[1]);
14         k = Integer.parseInt(s[2]);
15         visited = new boolean[n][m];
16         arr = new char[n][m];
17         int res = 0;
18         for(int i = 0; i < n; ++i){
19             arr[i] = cin.nextLine().toCharArray();
20         }
21         for(int x=0; x<n;++x){
22             for(int y =0;y<m;++y){
23                 if(visited[x][y] || arr[x][y] == '0') {
24                     continue;
25                 }
26                 if(bfs(x,y) < k){
27                     res += 1;
28                 }
29             }
30         }
31         System.out.println(res);
32     }
33     static int bfs(int x,int y){
34         visited[x][y] = true;
35         int ans = 0;
36         if(arr[x][y] == 'E'){
37             ans += 1;
38         }
39         LinkedList<int[]> q = new LinkedList<>();
40         q.add(new int[] {x,y});
41         while(q.size() > 0){
42             int [] tem = q.removeFirst();
43             int i = tem[0],j = tem[1];
44             for(int [] dir:dirs){
45                 int a = i+dir[0];
46                 int b = j + dir[1];
47                 if(a >= 0 && a < n && b >=0 && b < m && !visited[a][b] && arr[a][b] != '0'){
48                     visited[a][b] = true;
49                     if(arr[a][b] == 'E'){
50                         ans +=1;
51                     }
52                     q.add(new int[] {a,b});
53                 }
54             }
55         }
56         return ans;
57     }
58 }

```

