

#include<iostream>

#include<cstring>

#include<stdio.h>

#include<algorithm>

#include<queue>

using namespace std;

#define x first

#define y second

typedef pair<int,int> PII ;

const int Num = 40;

//本题先用bfs 来 实现 bfs实现后再用dfs来实现

char re[Num][Num];

char End[Num];

int juage[Num][Num];

int sum ;

int x ,y;

PII start;

void dfs(PII idx){

    //dfs先判断特殊的返回条件

     if(juage[idx.y][idx.x]== 1){

       return ;

    }

    //每层关键数据的处理

    juage[idx.y][idx.x]=1;

    sum++;

    int lx[]= {-1,1,0,0};

    int ly[]= {0,0,-1,1};

    for(int i = 0 ; i < 4 ; i++){//每层的遍历

      int nx = idx.x + lx[i],ny = idx.y + ly[i] ;

      if(nx < 0 || nx >= x || ny < 0 || ny >= y || re[ny][nx]=='#'){

            continue;

        }

    dfs(make\_pair(nx,ny));

    }

    return ;

}

int main(void){

      scanf("%d %d",&x , &y);

    while(x!=0 && y!=0){

    for(int i = 0 ; i < y ; i++){

        scanf("%s",re[i]);

    }

    for(int i = 0  ;i < y ; i++){

        for(int j = 0 ; j < x ; j++){

            if(re[i][j]=='@'){//得到bfs的起点坐标

                start.x = j;

                start.y = i ;

                break;

            }

        }

    }

    sum = 0;

    memset(juage,0,sizeof(juage));

    dfs(start);

    cout <<sum <<endl;

    scanf("%d %d",&x , &y);

    }

    return  0;

}