状压： 第2行一个正整数P，表示门和钥匙的数量；

#include <limits.h>using namespace std;const int maxn = 55;

char mp[maxn][maxn];

int min\_step[maxn][maxn][1<<10]; // 所到达的最小步数

int dir[4][2] = {0,1,0,-1,1,0,-1,0};

int n, m, key\_num;

int sx, sy, ex, ey;

/// queue< pair<int, int> > q;

struct Node{

int x, y, key\_door, step;

/// 坐标 钥匙\_门 当前步数};

bool bianjie(int x,int y{

if(x<=0 || y<=0 || x>n || y>m || mp[x][y]=='#')

return false;

return true;}

int bfs(){

queue <Node> q;

q.push({sx, sy, 0, 0});

memset(min\_step, 0, sizeof(min\_step));

while(!q.empty())

{

Node tmp = q.front();// 当前位置

q.pop();

int xx = tmp.x;

int yy = tmp.y;

for(int i = 0; i < 4; i++)

{

Node Next = tmp;// 下一个位置

int dx = xx + dir[i][0];

int dy = yy + dir[i][1];

int key\_door = Next.key\_door;

int step = Next.step+1;

if( !bianjie(dx, dy)) continue;

if( mp[dx][dy] != '.' && mp[dx][dy] != '&' && mp[dx][dy] != '$')

{

if(isupper(mp[dx][dy]))

{

int num = mp[dx][dy] - 'A';

if( !(key\_door&(1<<num)) ) continue;

/// 状压为0 没有拿到钥匙

else ;/// 门可以开 则可以

}

else if(islower(mp[dx][dy]))

{

int num = mp[dx][dy] - 'a';

key\_door = key\_door|(1<<num);

/// set设置 拿到钥匙

}

}

// 之前的最小步数 大于 当前过来的步数

if( min\_step[dx][dy][key\_door] > step

|| min\_step[dx][dy][key\_door] == 0 )

{

min\_step[dx][dy][key\_door] = step;// 更新最小步数

q.push(Node{dx, dy, key\_door, step});

}}}}

int main(){

scanf("%d %d", &n, &m);

scanf("%d", &key\_num);

for(int i = 1; i <= n; i++)

{

getchar();

for(int j = 1; j <= m; j++)

{

scanf("%c", &mp[i][j]);

if( mp[i][j] == '$')

sx = i, sy = j;

if( mp[i][j] == '&')

ex = i, ey = j;

}

}

bfs();

int ans = INT\_MAX;

for(int i = 0; i < (1<<key\_num) ; i++)

if( min\_step[ex][ey][i] != 0 &&

min\_step[ex][ey][i] < ans )

ans = min\_step[ex][ey][i];

printf("%d\n", ans==INT\_MAX?-1:ans);

}

5 5

1

&A..$

####. ...#. .#.#. a#...