Tempter of the Bone

The doggie found a bone in an ancient maze, which fascinated him a lot. However, when he picked it up, the maze began to shake, and the doggie could feel the ground sinking. He realized that the bone was a trap, and he tried desperately to get out of this maze.   
  
The maze was a rectangle with sizes N by M. There was a door in the maze. At the beginning, the door was closed and it would open at the T-th second for a short period of time (less than 1 second). Therefore the doggie had to arrive at the door on exactly the T-th second. In every second, he could move one block to one of the upper, lower, left and right neighboring blocks. Once he entered a block, the ground of this block would start to sink and disappear in the next second. He could not stay at one block for more than one second, nor could he move into a visited block. Can the poor doggie survive? Please help him.

**Input**

The input consists of multiple test cases. The first line of each test case contains three integers N, M, and T (1 < N, M < 7; 0 < T < 50), which denote the sizes of the maze and the time at which the door will open, respectively. The next N lines give the maze layout, with each line containing M characters. A character is one of the following:   
  
'X': a block of wall, which the doggie cannot enter;   
'S': the start point of the doggie;   
'D': the Door; or   
'.': an empty block.   
  
The input is terminated with three 0's. This test case is not to be processed.

**Output**

For each test case, print in one line "YES" if the doggie can survive, or "NO" otherwise.

**Sample Input**

4 4 5

S.X.

..X.

..XD

....

3 4 5

S.X.

..X.

...D

0 0 0

**Sample Output**

NO

YES

#include<stdio.h>

#include<string.h>

int flag,sx,sy,ex,ey,num;

int n,m,t,vis[10][10];

int dx[]={-1,0,1,0};

int dy[]={0,-1,0,1};

char map[10][10];

int abs(int p)

{

return p>=0?p:-p;

}

void dfs(int x,int y,int sum)

{

int i,xx,yy;

if(flag==1)

return;

if(x==ex&&y==ey&&sum==t)

{

flag=1;

return;

}

int mindis=abs(x-ex)+abs(y-ey); /\*当前点到终点的最短距离\*/

if(mindis>t-sum||(mindis+ t-sum )%2!=0)

return;

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

{

xx=x+dx[i];

yy=y+dy[i];

if(xx>=0&&xx<n&&yy>=0&&yy<m&&!vis[xx][yy]&&map[xx][yy]!='X') //没越界，点没走过，不是墙

{

vis[xx][yy]=1;

dfs(xx,yy,sum+1);

vis[xx][yy]=0;

}

}

}

int main()

{

int i,j;

while(~scanf("%d%d%d",&n,&m,&t))

{

if(n==0&&m==0&&t==0)

break;

num=0;

for(i=0;i<n;i++)

{

scanf("%s",map[i]);

for(j=0;j<m;j++)

{

if(map[i][j]=='S')

{

sx=i;

sy=j; /\*记录起点坐标\*/

}

if(map[i][j]=='D')

{

ex=i;

ey=j; /\*记录终点坐标\*/

}

if(map[i][j]=='X')

num++; /\*记录墙的数量\*/

}

}

if(n\*m-num-1<t)//n\*m是所有的点数，减去墙的数量就是能走的点数，

//而如果总共能走的点数还不够时间的话（就是说把能走的都走完了还没到时间，肯定就不行了

{

printf("NO\n");

continue;

}

flag = 0;

memset(vis,0,sizeof(vis));

vis[sx][sy] = 1;//记录一个点是否被走过，走过就不能再走了

dfs(sx,sy,0);

if(flag)

printf("YES\n");

else

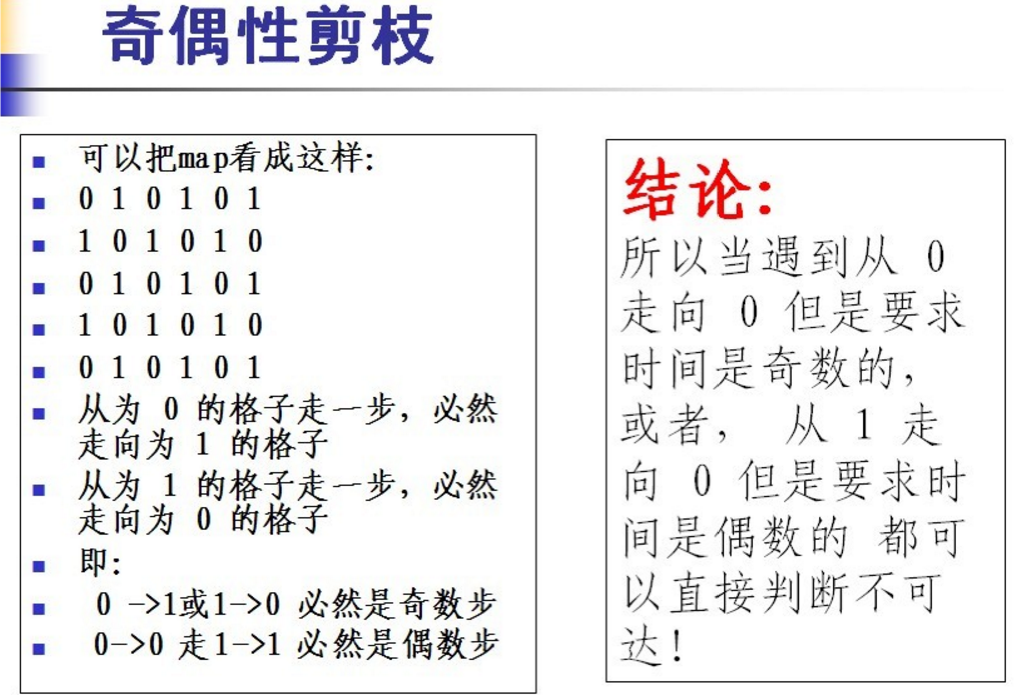
printf("NO\n");

}

return 0;

}

第一个剪枝我们可以想到，当剩下的步数大于剩下的时间的时候，狗是不能走到的；



再看这段代码

**[cpp]** [view plain](http://blog.csdn.net/libin56842/article/details/8962512) [copy](http://blog.csdn.net/libin56842/article/details/8962512)

1. tem = t-cnt - s1 - s2;

其中t-cnt味剩余的步数或者说时间，另其为T

s1+s2为剩余步数，另其为S

如果走偶数步要求的时间是奇数，或者走奇数步要求的时间是偶数，都明显不可行

而轻易得出奇数-偶数 = 奇数，反之亦然

而奇数-奇数= 偶数，偶数-偶数=偶数

所以tem必须为偶

奇偶剪枝：根据题目，doggie必须在第t秒到达门口。也就是需要走t-1步。设doggie开始的位置为（sx,sy）,目标位置为（ex,ey）.如果abs(ex-x)+abs(ey-y)为偶数，则abs(ex-x)和abs(ey-y)奇偶性相同，所以需要走偶数步；

当abs(ex-x)+abs(ey-y)为奇数时，则abs(ex-x)和abs(ey-y)奇偶性不同，到达目标位置就需要走奇数步。先判断奇偶性再搜索可以节省很多时间，不然的话容易超时。t-sum为到达目标位置还需要多少步。因为题目要求doggie必须在第t秒到达门的位置，所以（t-step）和abs(ex-x)+abs(ey-y)的奇偶性必然相同。因此temp=（t-step）-abs(ex-x)+abs(ey-y)必然为偶数。