学号：20152100121 班级：2班 姓名：林伟业 zoj用户名：clavier

题目一：zoj 1094

解题思路：从输入流读入x、y的坐标，然后根据坐标算出半径，再用半径算出半圆的面积，再用面积除50就可以得出结果。

程序清单：

#include<stdio.h>

#include<math.h>

#define PI 3.141592653

int main()

{

int m,n,i;

double b,c;

double x,y;

scanf("%d",&m);

for(i=1;i<=m;i++)

{

scanf("%lf %lf",&x,&y);

n=(x\*x+y\*y)\*PI/100;

b=50\*n-(x\*x+y\*y)\*PI/2;

c=50\*(n+1)-(x\*x+y\*y)\*PI/2;

if(b<0&&c>0)

n=n+1;

printf("Property %d: This property will begin eroding in year %d.\n",i,n);

}

printf("END OF OUTPUT.");

}

解题总结：题目属于简单题，能看懂题目加上会数学公式就可以直接做完。注意读入数据和输出的格式，变量全部用到double，不能用平常的int，程序多处用到PI，所以讲PI定义为常亮，减少错误，增加程序的可读性

题目二：zoj 1091

解题思路：使用bfs，知道开始位置的坐标和目标的坐标，求最段路径，需要用到队列，一开始从初始位置出发，将现在的位置坐标加到队列并标记此位置已经走过，再从队列弹出，得到坐标，从队列得到的坐标出发，向它8个方向走，将8个方向的坐标加入到队列，并标记位置已经走过，又从队列弹出得到队列头部的坐标，走到一个位置就将8个方向中还没有走过的坐标加人队列，递归地加人和弹出，知道走到目标的坐标。

程序清单：

#include <iostream>

#include <cstring>

#include <queue>

using namespace std;

const string display0 = "To get from ";

const string display1 = " to ";

const string display2 = " takes ";

const string display3 = " knight moves.";

int x[8] = {1,1,2,2,-1,-1,-2,-2};

int y[8] = {2,-2,1,-1,-2,2,-1,1};

bool board[8][8] =

{

0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0

};

class Knight

{

public:

int x;

int y;

int step;

Knight()

{

step = 0;

}

};

queue<Knight> q;

int bfs(Knight start,Knight end)

{

if (start.x == end.x && start.y == end.y)

return start.step;

for (int i = 0; i < 8; i++)

{

Knight now;

now.x = start.x + x[i];

now.y = start.y + y[i];

if (board[now.x][now.y] || now.x < 0 || now.y < 0 || now.x > 7 || now.y > 7)

continue;

now.step = start.step + 1;

board[now.x][now.y] = 1;

if (start.x == end.x && start.y == end.y)

return now.step;

q.push (now);

}

Knight first = q.front ();

q.pop ();

bfs (first,end);

}

int main()

{

string a;

string b;

while (cin >> a >> b)

{

while (!q.empty())

{

q.pop();

}

memset (board,0,sizeof (board));

int x = a[0] - 'a';

int y = a[1] - '1';

int xEnd = b[0] - 'a';

int yEnd = b[1] - '1';

Knight kStart;

Knight kEnd;

kStart.x = x;

kStart.y = y;

kStart.step = 0;

board[kStart.x][kStart.y] = 1;

kEnd.x = xEnd;

kEnd.y = yEnd;

int anw = bfs(kStart,kEnd);

cout << display0 << a << display1 << b << display2 << anw << display3 << endl;

}

return 0;

}

解题总结：读入数据时候是字符，首先压迫转换成坐标的整形类型，每个坐标是一个对象，所以定义队列的时候需要注意队列的类型，从坐标出发向8个方向出发，是否加入队列前要判断走到的这个点是否出界和是否走过，递归的部分是入队列和出队列，直到走到目标坐标。设计递归需要清楚重复的算法是那个。

题目三：zoj 2734

解题思路：目标是找出所有可能，所以采用穷举遍厉价值相加的所有可能，当价值等于期望价值时，兑换方式加1，当数量为零或总价值比期望的大的时候就停止遍厉。使用一个函数实现加载求和，并递归调用此函数求出最后价值。最后输出兑换的方式。

程序清单：

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <string.h>

using namespace std;

int sum,ways;

int num[1010];

int value,n;

void DFS(int x)

{

if( sum == value )

{

ways++;

return ;

}

for(int i=x; i<=value; i++)

{

if( num[i] && sum + i <= value )

{

num[i]--;

sum += i;

DFS(i);

num[i]++;

sum -= i;

}

}

}

int main(void)

{

int x,y;

int pe = 0;

while( cin >> value >> n )

{

if( pe )

cout << endl;

pe = 1;

memset(num,0,sizeof(num));

sum = ways = 0;

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

{

cin >> x >> y;

num[x] = y;

}

DFS(1);

cout << ways << endl;

}

return 0;

}

解题总结：需要巧妙利用递归函数，知道重复算法在那里。程序中使用数组下标表示卡片的价值，数组下标对应的的值表示卡片的数量，一一对应的关系，使用更方便。