14级第六次上机解题报告

14211135寇宇增

第六题

这个题做出来也可能算是个巧合吧，看到这题的时候我们都知道这个是用迭代迭出来的迭代三角形，但是想不到如何进行迭代。然后我就随手画了画图，突然发现一个差不多的东西，它就是杨辉——帕斯卡三角，相同点在哪里呢，就是我们可以根据我们要画的迭代三角形往它身上凑，结果发现了一个小规律：把所有的奇数变成/\，偶数变成两个空格，得到的结果竟然就是这个迭代三角形！这是一个很神奇的事情。但是我们要是用杨辉——帕斯卡三角作为基础去换的话，可能会出现的一种情况就是——爆了int了，或者说，没有一个可以画更长远的做法。所以我们用奇偶检验的思想，把奇数赋值成1，偶数赋值成0，然后我们这样就可以把变量定成bool型，而且绝对不会爆的。再然后，我们根据奇偶运算，发现其实变成我们离散中学过的东西，就叫做抑或，用^表示。所以呢，我们就可以画画啦。

先是用一个二维数组保存好我们的数据，就是在内存里画出来一个10表示的三角，然后，我们再画成题目中让画的三角就可以啦。

程序代码如下：

#include <cstdio>

int main()

{

int n;

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

{

bool z[ n ] [ n ];

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

{

for( int b = 1; b < a + 1; b ++)

{

if ( b == 1 || b == a)

z [ a - 1 ] [ b - 1]= 1;

else

{

bool t = z [ a - 2] [ b - 2] ^z [ a - 2] [ b - 1];

z [ a - 1] [ b - 1] = t;

}

}

}

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

{

for ( int num = 0; num < n - a; num ++)

{

printf(" ");

}

for ( int b = 1; b < a + 1; b ++)

{

if ( z [ a - 1] [ b - 1] )

printf("/\\");

else

printf(" ");

}

printf("\n");

}

}

return 0;

}

第七题

这个题做的实在是有点微软的风格，为什么呢，因为我打补丁了，哈哈。首先，得感谢于济凡同学做的比对程序，才有机会有效率的找到要打补丁的地方。怎么说呢，这个题用标准的方法会超时的，标准的方法慢的地方在年数判读的时候循环太多了，所以我们要想办法减少循环。

减少年份循环的办法：由于闰年的判断和400，100，4这三个数有关，所以我们就四百四百的减然后一百一百的减，然后四四的减，再一一的减，这样就快多了。减去相应的那么多年就行了。做出来比对一下，发现错的天数是有规律的，所以我们打了第一个补丁，叫做service pack 1，然后，我们又发现一些有规律的错误，就打了service pack 2这个补丁。当然，我和济凡同学还基于他的算法研究出来了更漂亮的没有补丁的代码，但是时间却慢一小些，详细的请见于济凡同学的解题报告。对了，还有一个两位数输出的小技巧，是%02d，这样就不用再判断是不是要前面加零了。

程序代码如下：

#include <cstdio>

//#include<iostream>

//using namespace std;

bool run ( int );

void week( int );

int month [ 2 ] [ 13 ] = { { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365} , { -1, 30, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365} } ;

int main()

{

int n;

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

{

int save = n;

int m = 12, d = 31, y = 0 ;

int step [ 4 ];

step [ 0] = n / ( 365 \* 400 + 97 );

y += step [ 0 ]\* 400;

n -= ( 365 \* 400 + 97 ) \* step [ 0 ]; //小于400年

//cout << step [ 0 ] <<" " << y <<" "<< n <<endl;

step [ 1 ]= n / (365 \* 100 + 25);

y += step [ 1 ] \* 100;

n -= ( 365 \* 100 + 24 ) \*step [ 1 ]; //小于100年

//cout << step [ 1 ] <<" " << y <<" "<< n <<endl;

step [ 2 ]= n / ( 365 \* 4 + 1 );

y += step [ 2 ] \* 4;

n -= ( 365 \* 4 + 1 ) \* step[ 2 ];//小于4年

//cout << step [ 2 ] <<" " << y <<" "<< n <<endl;

step [ 3 ]= n / 365;

y += step[ 3 ];

n -= 365 \* step [ 3 ];//小于一年

//cout << step [ 3 ] <<" " << y <<" "<< n <<endl;

y += 2000;

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

{

bool jus = run( y );

if( month [ jus ] [ num ] < n && month [ jus ] [ num + 1 ] >= n)

{

d = n - month [ jus ] [ num ];

m = num + 1;

break;

}

}

//service pack 1

if( n == 0 )

{

y --;

if( y % 4 == 3 && step [ 3 ] == 0 )

{

m = 1;

d = 1;

y++;

}

}

//service pack 2

if ( y % 400 == 300 && n == 1)

d ++;

printf("%d-%02d-%02d " , y, m, d );

week( save );

}

return 0;

}

bool run ( int n )

{

if( (n % 4 == 0 && n % 100 != 0) || n % 400 == 0)

return 1;

else

return 0;

}

void week( int n )

{

int m = n % 7;

switch (m)

{

case 0:

printf("Saturday\n");

break;

case 1:

printf("Sunday\n");

break;

case 2:

printf("Monday\n");

break;

case 3:

printf("Tuesday\n");

break;

case 4:

printf("Wednesday\n");

break;

case 5:

printf("Thursday\n");

break;

case 6:

printf("Friday\n");

break;

default:

printf("Wrong\n");

break;

}

}