**算法训练 黑色星期五**

时间限制：1.0s   内存限制：512.0MB

**问题描述**  
　　有些西方人比较迷信，如果某个月的13号正好是星期五，他们就会觉得不太吉利，用古人的说法，就是“诸事不宜”。请你编写一个程序，统计出在某个特定的年份中，出现了多少次既是13号又是星期五的情形，以帮助你的迷信朋友解决难题。  
　　说明：（1）一年有365天，闰年有366天，所谓闰年，即能被4整除且不能被100整除的年份，或是既能被100整除也能被400整除的年份；（2）已知1998年1月1日是星期四，用户输入的年份肯定大于或等于1998年。  
　　输入格式：输入只有一行，即某个特定的年份（大于或等于1998年）。  
　　输出格式：输出只有一行，即在这一年中，出现了多少次既是13号又是星期五的情形。  
**输入输出样例**

样例输入

1998

样例输出

3

本题的C++参考代码如下：

#include<iostream>

using namespace std;

//判断闰年

bool fun1(int n)

{

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

return true;

return false;

}

//判断输入年份第一天为星期几

int fun2(int n)

{

if (n==1998)

{

return 4;

}

else

{

int sum=0;

for (int i=1998;i<n;i++)

{

if(fun1(i))

sum+=366;

else

sum+=365;

}

return (sum+4)%7;

}

}

//计算黑色星期五

int fun3(int year,int fx)

{

int i=0;

int a[12]={12,43,71,102,132,163,193,224,255,285,316,346};

int b[12]={12,43,72,103,133,164,194,225,256,286,317,347};

if (fun1(year))

{

for (int j=0;j<12;j++)

{

if ((b[j]+fx)%7==5)

i++;

}

}

else

{

for (int j=0;j<12;j++)

{

if ((a[j]+fx)%7==5)

i++;

}

}

return i;

}

int main()

{

int n;

cin>>n;

cout<<fun3(n,fun2(n))<<endl;

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

int getMonthNumber(int month)

{

switch(month)

{

case 1:return 31;

case 2:return 28;

case 3:return 31;

case 4:return 30;

case 5:return 31;

case 6:return 30;

case 7:return 31;

case 8:return 31;

case 9:return 30;

case 10:return 31;

case 11:return 30;

case 12:return 31;

}

}

int isrunnian(int year)

{

if(year%400==0 || (year%100!=0 && year%4==0)) return 1;

return 0;

}

int main()

{

int year,i,sumday=0,begin,day,month,j,time=0;

scanf("%d",&year);

for(i=1998;i<year;i++)

sumday+=isrunnian(i)==1?366:365;

sumday%=7;

switch(sumday)

{

case 0:begin=4;break;

case 1:begin=5;break;

case 2:begin=6;break;

case 3:begin=7;break;

case 4:begin=1;break;

case 5:begin=2;break;

case 6:begin=3;break;

}

//计算出year年的一月1号为星期begin

for(i=1;i<=12;i++)

{

if(i==2 && isrunnian(year)==1) day=29;

else day=getMonthNumber(i);

for(j=1;j<=day;j++)

{

if(begin==5 && j==13) time++;

begin++;

if(begin==8) begin=1;

}

}

printf("%d",time);

}

本题的Java参考代码如下：

import java.util.Scanner;

public class Main {

public static boolean runnian(int year)

{

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

return true;

else return false;

}

public static int total(int year)

{

int total=0;

int years=0;

int count=0;

if(year<2000)

years=(year-1998)\*365;

else

years=(((year-2000)/4)+1)\*366+(year-1998-(((year-2000)/4)+1))\*365;

if(runnian(year))

years=((year-2000)/4)\*366+(year-1998-((year-2000)/4))\*365;

int mouth []=new int []{0,31,0,31,30,31,30,31,31,30,31,30,31};

if(runnian(year))

mouth[2]=29;

else mouth[2]=28;

for(int i=1;i<=12;i++)

{

if((years+mouth[i-1]+13-4)%7==5)

{

count++;

}

years=years+mouth[i-1];

}

return count;

}

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

int year=in.nextInt();

int i=total(year);

System.out.println(i);

}

}