#include <stdio.h>

#include <stdlib.h>

//日期结构体

typedef struct D

{

int year;

int month;

int day;

}Date;

int getDays(Date date1,Date date2);//获取两个日期间的天数

int IsLeapYear(int year);//是否是闰年

int GetMaxDay(int year,int month);//获取某个月的最大天数

int main(int argc,char\*\* argv)//主函数

{

int days;

Date start,end;//start:开始日期，end：结束日期

printf("请输入第一个日期（1949.10.1）：");

scanf("%d.%d.%d",&start.year,&start.month,&start.day);//录入起始日期

printf("请输入第二个日期（1959.10.1）：");

scanf("%d.%d.%d",&end.year,&end.month,&end.day);//录入结束日期

days=getDays(start,end);//调用函数获取两个日期间相差的天数

printf("两个日期间隔着%d天\n",days);//输出相差的天数

system("pause");

return 0;

}

//判断是否是闰年

int IsLeapYear(int year)

{

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

//（年份是否能被400整除）或（年份能被4但不能被100整除））

}

int GetMaxDay(int year,int month)//获得某年某月的最大天数

{

switch(month)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

return 31;//这几月返回31

case 4:

case 6:

case 9:

case 11:

return 30;//这几月返回30

case 2:

return IsLeapYear(year)?29:28;//根据闰年判断结果计算2月天数

default:

return -1;

}

}

int getDays(Date date1,Date date2)

{ int days=0;//预留中间求和使用

if(date1.year==date2.year && date1.month==date2.month && date1.day==date2.day)

return 0;//若是同一天，自然返回0

else

{

if(date1.year==date2.year && date1.month==date2.month)//同一年同一个月中的两天

{

days=date2.day-date1.day;

return days;//用末尾天减去起始天

}

else if(date1.year=date2.year)//同一年不同月

{

int start\_maxday=GetMaxDay(date1.year,date1.month);//

days = start\_maxday-date1.day;//当月的最大天数-起始天数

if(date2.month==date1.month+1)//若末尾月即为下一月

{

days+=date2.day;//刚刚的天数加上末尾月的天数即可

return days;

}

else

{

int m=date2.month,n=date1.month+1;

if(n!=m)

{

days+=GetMaxDay(date1.year,n);//从起始月的下一月开始的每一月最大天数之和

n++;

}

days+=date2.day;//加上末尾月的天数

return days;

}

}

}

}