1 从键盘上输入字符，将小写字母转换成大写字母。输入“ctl + z” 结束

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main() {

char ch;

while (fflush(stdin), scanf("%c", &ch) != EOF) {

printf("%c\n", ch - 'a' + 'A'); // -32

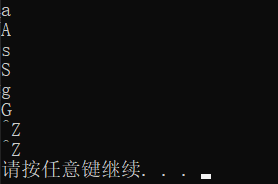
scanf("%c", &ch);

}

system("pause");

return 0;

}



3.1 将十进制数转换成二进制数

int main()

{

int n;

while (fflush(stdin), scanf("%d", &n) != EOF)

{

int i, j, m = -1, a[16]; //m 为n的长度计数器

while (n != 0) {

i = n % 2;

a[++m] = i;

n = n / 2;

}

for (j = m; j >= 0; j--) {

printf("%d", a[j]);

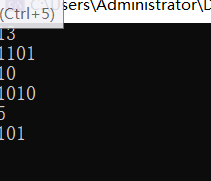
}

printf("\b\n");

}

return 0;

}



3.2 将二进制转换成十进制数

int main() // 2.2： 2--》10

{

char num[30];

while (fflush(stdin), scanf("%s", &num) != EOF)

{

int n = 0;

for (int i = 0; num[i] != '\0'; ++i) {

n \*= 2;

n += num[i] - '0';

}

printf("%d\n", n);

}

return 0;

}



3.3 10--》16

int main() // 2.3： 10--》16

{

int n;

while (fflush(stdin), scanf("%d", &n) != EOF)

{

char a[30];

int i, j, m = -1; //m 为n的长度计数器

int temp;

while (n != 0) {

i = (n%16 < 10)? (n%16 +'0'):(( n%16 -10)+'a');

a[++m] = i;

n = n / 16;

}

for (j = m; j >= 0; j--) {

printf("%c", a[j]);

}

printf("\b\n");

}

return 0;

}



3.4 16--》10

int main()

{

char c[10];

int i, j, sum = 0;

printf("请输入一个十六进制数:");

while (fflush(stdin), scanf("%s", &c) != EOF) {

//scanf("%s", c);

for (i = strlen(c) - 1, j = 0; i >= 0; i--, j++)

{

if (isdigit(c[i]))

sum += ((int)c[i] - 48) \* pow(16, j);

if (isalpha(c[i]))

if (c[i] <= 'Z') {

sum += ((int)c[i] - 55) \* pow(16, j);

}

else {

sum += ((int)c[i] - 87) \* pow(16, j);

}

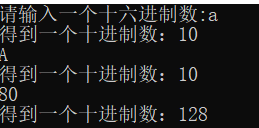
}

printf("得到一个十进制数：%d\n", sum);

sum = 0;

}

}



4 统计一个整数对应的二进制数的1的个数

int main()

{

unsigned int s = 1 << 31; //s = 1000 0000 0000 0000 0000 0000 0000 0000

int n, count = 0;

printf("请输入一个整数： ");

while (fflush(stdin), scanf("%d", &n) != EOF)

{

if (n < 0) {

n = ~n + 1; //求补码

n = n | s;

}

for (int i = 0; i < 32; i++) {

if (n & s) {

count++;

}

s >>= 1;

}

s = 1 << 31;

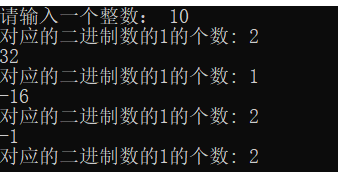
printf("对应的二进制数的1的个数: %d\n", count);

count = 0;

}

return 0;

}



6.1 输入年月日，输出该日期是当年的第几天。

int run(int year)

{

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

return 2; // 闰年

}

else {

return 1; // 非闰年

}

}

int cal(int year, int month)

{

int x, i, s = 0;

int date[2][12] =

{

{31,28,31,30,31,30,31,31,30,31,30,31},

{31,29,31,30,31,30,31,31,30,31,30,31}

};

x = run(year);

for (i = 1; i < month; i++) {

s += date[x - 1][i];

}

return s;

}

//int DayOfYear() {

//

//}

int main() //5. 统计一个整数对应的二进制数的1的个数

{

int year,month,day,tian;

printf("输入年月日，空格隔开： ");

while (fflush(stdin), scanf("%d%d%d", &year, &month, &day) != EOF)

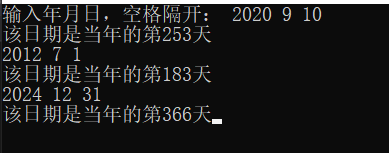
{

tian = day + cal(year,month);

printf("该日期是当年的第%d天", tian);

}

}



6.2 输入两个日期（年 月 日 年 月 日）， 输出这两个日期之间差多少天。

int run(int year) //输入两个日期（年 月 日 年 月 日）， 输出这两个日期之间差多少天

{

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

return 2; // 闰年

}

else {

return 1; // 非闰年

}

}

int cal(int year, int month)

{

int x, i, s = 0;

int date[2][12] =

{

{31,28,31,30,31,30,31,31,30,31,30,31},

{31,29,31,30,31,30,31,31,30,31,30,31}

};

x = run(year);

for (i = 1; i < month; i++) {

s += date[x - 1][i];

}

return s;

}

int nianSum(int year, int year2) {

int sum = 0;

for (int i = year; i < year2; i++) {

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

sum += 366; // 闰年

}

else {

sum += 365;

}

}

return sum;

}

int main()

{

int year, month, day, tian;

int year2, month2, day2, tian2;

int cha;

while (fflush(stdin), scanf("%d%d%d%d%d%d", &year, &month, &day, &year2, &month2, &day2) != EOF)

{

tian = day + cal(year, month);

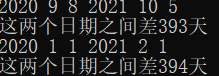
tian2 = day2 + cal(year2, month2);

cha = tian2 - tian;

printf("这两个日期之间差%d天", cha + nianSum(year, year2));

}

}



6.3 输入一个日期，输出该日期是星期几

int run(int year) //输入两个日期（年 月 日 年 月 日）， 输出这两个日期之间差多少天

{

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

return 2; // 闰年

}

else {

return 1; // 非闰年

}

}

int cal(int year, int month)

{

int x, i, s = 0;

int date[2][12] =

{

{31,28,31,30,31,30,31,31,30,31,30,31},

{31,29,31,30,31,30,31,31,30,31,30,31}

};

x = run(year);

for (i = 1; i < month; i++) {

s += date[x - 1][i];

}

return s;

}

int nianSum(int year, int year2) {

int sum = 0;

for (int i = year; i < year2; i++) {

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

sum += 366; // 闰年

}

else {

sum += 365;

}

}

return sum;

}

int main() //5. 统计一个整数对应的二进制数的1的个数

{

int year=1, month=1, day=1, tian;

int year2, month2, day2, tian2;

int cha;

while (fflush(stdin), scanf("%d%d%d", &year2, &month2, &day2) != EOF)

{

tian = day + cal(year, month);

tian2 = day2 + cal(year2, month2);

cha = tian2 - tian +nianSum(year, year2);

printf("公园初距今%d天", cha + nianSum(year, year2));

printf("这天是星期%d", 1+(cha % 7));

}

}



6.4 输入 一个日期 和一个整数 n，输出从该日期起经过n天以后的日期

int isLeap(int year)

{

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

{

return 1;

}

else

{

return 0;

}

}

int date\_cacl(int year, int mon, int day, int n)

{

int a[13] = { 31,28,31,30,31,30,31,31,30,31,30,31,0 };//a[13]用于判断是否跨年

int i;

int tmp1 = 0, sum = 0, tmp2 = 0;//tmp1用于记录天数，tmp2用于记录年份

//step1：计算输入日期距离年初的天数

for (i = 0; i < mon - 1; i++)

{

tmp1 += a[i];

}

tmp1 += day;

if (mon > 2 && isLeap(year))

{

tmp1 += 1;

}

tmp1 += n;//距离年初的总天数

// step2 temp1中包含多少个月，是否跨年，减整月后剩下多少天

for (i = 1; tmp1 > a[i]; i++) {

tmp1 -= a[i];

if (2 == i && isLeap(year)) {

tmp1 -= 1;

}

if (a[i] == 0) {

i = 1;

year += 1;

}

}

printf("经过%d天后的日期是%d-%d-%d\n", n, year, i, tmp1);

return 0;

}

int main()

{

int year, mon, day;

int n;

printf("请输入一个日期和一个整数n:\n");

scanf("%d%d%d%d", &year, &mon, &day, &n);

date\_cacl(year, mon, day, n);

system("pause");

return 0;

}



6.5 输入一个年份,月份，输出该月份的日历。

int main()

{

int m, d, monthmax, y;

scanf("%d%d", &y, &m);

if (0 == y % 4 && y % 100 || 0 == y % 400) {

if ((m == 1) || (m == 3) || (m == 5) || (m == 7) || (m == 8) || (m == 10) || (m == 12)) {

monthmax = 31;

}

if (m == 2)

monthmax = 29;

if ((m == 4) || (m == 6) || (m == 9) || (m == 11))

monthmax = 30;

}

else {

if ((m == 1) || (m == 3) || (m == 5) || (m == 7) || (m == 8) || (m == 10) || (m == 12))

monthmax = 31;

if (m == 2)

monthmax = 28;

if ((m == 4) || (m == 6) || (m == 9) || (m == 11))

monthmax = 30;

}

//基姆拉尔森计算公式取得一号周几

if ((m == 1) || (m == 2))

{

m += 12; //一月算做上一年的十三月，二月算做上一年的十四月

y--;

}

d = (2 + 2 \* m + 3 \* (m + 1) / 5 + y + y / 4 - y / 100 + y / 400) % 7; //算出这月一号周几

if (d == 0)

{

d = 7;

}

// printf

printf(" 1 2 3 4 5 6 7 \n");

for (int i = 1; i < d ; i++) printf(" ");

for (int j = 1; j <= monthmax; j++) {

if (j < 10) printf(" %d ", j);

if (j >= 10) printf(" %d ", j);

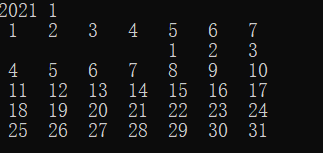
if ((j + d - 1) % 7 == 0) printf("\n");

if(j== monthmax) printf("\n");

}

printf("\n");

}



7 打印一年日历

void month\_table(int y, int m) {

int d, monthmax;

if (0 == y % 4 && y % 100 || 0 == y % 400) {

if ((m == 1) || (m == 3) || (m == 5) || (m == 7) || (m == 8) || (m == 10) || (m == 12)) {

monthmax = 31;

}

if (m == 2)

monthmax = 29;

if ((m == 4) || (m == 6) || (m == 9) || (m == 11))

monthmax = 30;

}

else {

if ((m == 1) || (m == 3) || (m == 5) || (m == 7) || (m == 8) || (m == 10) || (m == 12))

monthmax = 31;

if (m == 2)

monthmax = 28;

if ((m == 4) || (m == 6) || (m == 9) || (m == 11))

monthmax = 30;

}

//基姆拉尔森计算公式取得一号周几

if ((m == 1) || (m == 2))

{

m += 12; //一月算做上一年的十三月，二月算做上一年的十四月

y--;

}

d = (2 + 2 \* m + 3 \* (m + 1) / 5 + y + y / 4 - y / 100 + y / 400) % 7; //算出这月一号周几

if (d == 0)

{

d = 7;

}

// printf

printf(" 1 2 3 4 5 6 7 \n");

printf("=============================\n");

for (int i = 1; i < d; i++) printf(" ");

for (int j = 1; j <= monthmax; j++) {

if (j < 10) printf(" %d ", j);

if (j >= 10) printf(" %d ", j);

if ((j + d - 1) % 7 == 0) printf("\n");

if (j == monthmax) printf("\n");

}

printf("\n\n");

}

int main()

{

int m, y;

scanf("%d", &y);

for (int m = 1; m < 13; m++) {

printf(" %d月的月历表:\n", m);

month\_table(y, m);

}

}

