1.

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

int main()

{

char ch;

scanf\_s("%c\n", &ch, sizeof(ch));

printf("%c\n", toupper(ch));

return 0;

}

2.

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

void number(int n) {

printf(" %3d \t", n);

}

void asterisk() {

printf("\*\*\*\*\*\t");

}

void blank() {

printf(" \t");

}

int nmax(int a, int b, int c) {

return a > b ? (a > c ? a : c) : (b > c ? b : c);

}

void names() {

printf(" alp \t");

printf(" num \t");

printf(" oth \n");

}

int main()

{

int alp = 0, num = 0, oth = 0;

char ch;

while (scanf\_s("%c", &ch, sizeof(ch))) {

if (isalpha(ch))

++alp;

else if (isdigit(ch))

++num;

else

++oth;

}

int max = nmax(alp, num, oth);

int alpnm = 1, numnm = 1, othnm = 1;//控制数字的输出

for (int i = max; i >= 0; --i) {

if (alpnm && alp == i) {//第一个柱子

number(alp);

--alpnm;

--alp;

}else if (alp == i) {

asterisk();

--alp;

}

else

blank();

if (numnm && num == i) {//第二个柱子

number(num);

--numnm;

--num;

}

else if (num == i) {

asterisk();

--num;

}

else

blank();

if (othnm && oth == i) {//第三个柱子

number(oth);

--othnm;

--oth;

}

else if (oth == i) {

asterisk();

--oth;

}

else

blank();

printf("\n");//每一行换行

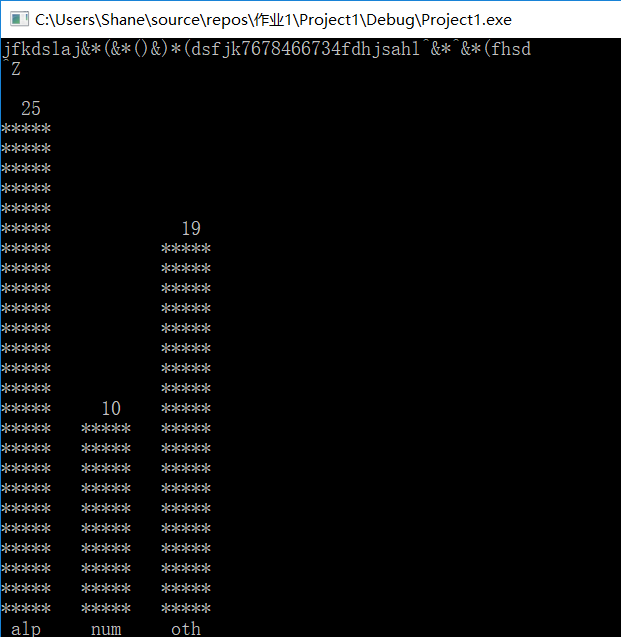
}

names();

system("pause");

return 0;

}



3.

（1）

//十进制转二进制

#include <stdio.h>

#include <stdlib.h>

void detobi(int n) {

char bina[50];

int i = 0;

while (n != 0) {

bina[i] = '0' + n % 2;

n /= 2;

++i;

}

while (i >= 0) {

printf("%c", bina[i-1]);

--i;

}

putchar('\n');

}

int main()

{

int num;

while (scanf\_s("%d", &num)) {

detobi(num);

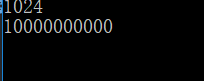
rewind(stdin);

}

system("pause");

return 0;

}



(2)

//二进制转十进制

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

int bitode(char\* pc, int n) {

int num = 0;

for (int i = 1; n > 0; --n) {

num = num + (pc[n] - '0') \* i;

i = 2 \* i;

}

return num;

}

int main()

{

char arch[50], ch;

int cnt = 0;

while (scanf\_s("%c", &ch, sizeof(ch))) {//以字符方式读入二进制数值，放于字符数组

if (isdigit(ch)) {

++cnt;

arch[cnt] = ch;

}

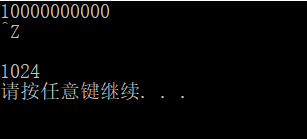
}

printf("%d\n", bitode(arch, cnt));

system("pause");

return 0;

}



(3)

//十转十六

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

#include <stdio.h>

#include <stdlib.h>

void detohex(int n) {

char bina[50];

int i = 0;

while (n != 0) {

int hex = n % 16;

if (hex >= 0 && hex <= 9)

bina[i] = '0' + hex;

else

bina[i] = 'a' + hex - 10;

++i;

n /= 16;

}

while (i >= 0) {

printf("%c", bina[i-1]);

--i;

}

putchar('\n');

}

int main()

{

int num;

while (scanf\_s("%d", &num)) {

detohex(num);

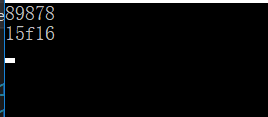
rewind(stdin);

}

system("pause");

return 0;

}



(4)

//十六转十

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

int hextode(char\* pc, int n) {

int num = 0;

for (int i = 1; n > 0; --n) {

if (isdigit(pc[n])) {//分别处理数字与字母部分

num = num + (pc[n] - '0') \* i;

}

else {

num = num + (pc[n] - 'a' + 10) \* i;

}

i = 16 \* i;

}

return num;

}

int main()

{

char arch[50], ch;

int cnt = 0;

while (scanf\_s("%c", &ch, sizeof(ch))) {//以字符方式读入十六进制数值，放于字符数组

if (isdigit(ch) ||

(tolower(ch) >= 'a' && tolower(ch) <= 'f')) {//只读入相关字符

++cnt;

arch[cnt] = ch;

}

}

printf("%d\n", hextode(arch, cnt));

system("pause");

return 0;

}

z 
1027653 

4.

//统计整数对应二进制中1的个数

#include <stdio.h>

#include <stdlib.h>

int count1(int n) {

int is1;

int cnt1 = 0;//计1的个数

while (n != 0) {

is1 = n % 2;

if (is1 == 1)//判断是否为1

++cnt1;

n /= 2;

}

return cnt1;

}

int main()

{

int num;

while (scanf\_s("%d", &num)) {

printf("%d\n",count1(num));

rewind(stdin);//刷新

}

system("pause");

return 0;

}

789789 
-z 
-z 

5.

（1）（2） （3）

//分别找出1001，1002，1003个整数中一个至三个非成对出现的数字

#include <stdio.h>

#include <stdlib.h>

#define sz 103

int ct1 = 0, ct2 = 0;//记录分组数组中所存的元素多少

int t = sz; //用于记录两个不同元素存放在一起的数组元素的多少

int findone(const int \*pa, int n)//处理含有一个不同元素的情况

{

int onenm = 0;

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

onenm ^= pa[i];//消除其他所有成对的整数

}

return onenm;

}

int fnd\_bit1(int nm, int x)//寻找右起第一个bit为1的位置

{

int r = 0;

if (x > 0) {

r = x + 1;

nm = nm >> r;

}

while (!(nm % 2)) {

nm = nm >> 1;

++r;

}

return r;

}

void group(const int \*pa, int n, int \*a1, int \*a2, int r)//将元素进行分组

{

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

if ((pa[i] >> r) % 2) {

a1[ct1] = pa[i];

++ct1;

/\*printf("\*\*\*\n");

printf("%d\n", ct1);\*/

}

else {

a2[ct2] = pa[i];

++ct2;

/\*printf("+++\n");

printf("%d\n", ct2);\*/

}

}

}

void prnt(int \*par, int size)//将三元素化解为一元素状况

{

int aaa[sz];

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

if (i < size)

aaa[i] = par[i];

else

aaa[i] = 0;

}

int total = findone(aaa, t);

int ct = 0;

ct = fnd\_bit1(total, ct);

int a1[sz], a2[sz];

for (int i = 0; i < sz; ++i) {//初始化为0

a1[i] = 0; a2[i] = 0;

}

group(aaa, t, a1, a2, ct);

if (ct1 % 2 && ct2 % 2) {

printf("%d\n%d\n", findone(a1, ct1), findone(a2, ct2));

}

else {

if (ct1 % 2) {

printf("%d\n", findone(a1, ct1));

t = ct2;

ct1 = 0; ct2 = 0;

prnt(a2, t);//分解为两元素的递归调用

}

else {

printf("%d\n", findone(a2, ct1));

t = ct1;

ct1 = 0; ct2 = 0;

prnt(a1, t);//分解为两元素的递归调用

}

}

}

int main()

{

int iarr[] = { 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,

26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50, 333,

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25, 888,

26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50, 666 };

prnt(iarr, sz);

/\*for (int i = 1; i < 51; ++i) {

printf(",%d", i);

}\*/

system("pause");

return 0;

}

333 
请 按 任 意 键 继 续 

6.

(1)

//年月日转天数

#include <stdio.h>

#include <stdlib.h>

#define leaps (years % 4 == 0) && (years % 100 != 0) || (years % 400 == 0)

int days(int years, int month, int day)

{

switch (month - 1) {

case 11:

day += 30;

case 10:

day += 31;

case 9:

day += 30;

case 8:

day += 31;

case 7:

day += 31;

case 6:

day += 30;

case 5:

day += 31;

case 4:

day += 30;

case 3:

day += 31;

case 2:

day += (leaps) ? 29 : 28;

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

case 1:

day += 31;

case 0:

break;

default:

printf("error!\n");

}

return day;

}

int main()

{

int yr, mon, dy;

while (scanf\_s("%d %d %d", &yr, &mon, &dy)) {

printf("years:%d month:%d day:%d\n", yr, mon, dy);

printf("The days is %d.\n", days(yr, mon, dy));

rewind(stdin);

}

system("pause");

return 0;

}

2019 3 27 
ears:2019 month:3 day:27 
he days is 86. 

(2)

//日期之差

#include <stdio.h>

#include <stdlib.h>

#define leaps (years % 4 == 0) && (years % 100 != 0) || (years % 400 == 0)

int day\_of\_year(int years, int month, int day)

{

switch (month - 1) {

case 11:

day += 30;

case 10:

day += 31;

case 9:

day += 30;

case 8:

day += 31;

case 7:

day += 31;

case 6:

day += 30;

case 5:

day += 31;

case 4:

day += 30;

case 3:

day += 31;

case 2:

day += (leaps) ? 29 : 28;

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

case 1:

day += 31;

case 0:

break;

default:

printf("error!\n");

}

return day;

}

int delt(int years, int year2)

{

int days = 0;

for (; years < year2; ++years) {

days += (leaps) ? 366 : 365;

}

return days;

}

int main()

{

int yr1, mon1, dy1;

int yr2, mon2, dy2;

int diff;

while (scanf\_s("%d %d %d %d %d %d", &yr1, &mon1, &dy1, &yr2, &mon2, &dy2)) {

printf("date1:%d %d %d\n", yr1, mon1, dy1);

printf("date2:%d %d %d\n", yr2, mon2, dy2);

diff = delt(yr1, yr2) - day\_of\_year(yr1, mon1, dy1) + day\_of\_year(yr2, mon2, dy2);

printf("The differences of the two dates is %d.\n", diff);

rewind(stdin);

}

system("pause");

return 0;

}

1995 5 22 2019 3 27 
atel:1995 5 22 
ate2:2019 3 27 
he differences of the two dates is 8710. 

（3）

//日期求星期

#include <stdio.h>

#include <stdlib.h>

#define leaps (years % 4 == 0) && (years % 100 != 0) || (years % 400 == 0)

#define base\_yr 1990

#define base\_mn 1

#define base\_dy 1

int day\_of\_year(int years, int month, int day)

{

switch (month - 1) {

case 11:

day += 30;

case 10:

day += 31;

case 9:

day += 30;

case 8:

day += 31;

case 7:

day += 31;

case 6:

day += 30;

case 5:

day += 31;

case 4:

day += 30;

case 3:

day += 31;

case 2:

day += (leaps) ? 29 : 28;

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

case 1:

day += 31;

case 0:

break;

default:

printf("error!\n");

}

return day;

}

int day\_dif\_year(int years, int year2)

{

int days = 0;

for (; years < year2; ++years) {

days += (leaps) ? 366 : 365;

}

return days;

}

void daytowk(int d)

{

switch (d % 7) {

case 0:

puts("Monday");

break;

case 1:

puts("Thusday");

break;

case 2:

puts("Wenesday");

break;

case 3:

puts("Thursday");

break;

case 4:

puts("Friday");

break;

case 5:

puts("Saturday");

break;

case 6:

puts("Sunday");

break;

}

}

int main()

{

int yr, mon, dy;

int diff;

while (scanf\_s("%d %d %d", &yr, &mon, &dy)) {

printf("date1:%d %d %d\n", yr, mon, dy);

diff = day\_dif\_year(base\_yr, yr) - day\_of\_year(base\_yr, base\_mn, base\_dy) + day\_of\_year(yr, mon, dy);

daytowk(diff);

putchar('\n');

rewind(stdin);

}

system("pause");

return 0;

}

2019 3 27 
date1:2019 3 27 
enesday 
-z 
-z 

（4）

//给日期转n天后日期

#include <stdio.h>

#include <stdlib.h>

#define leaps (year % 4 == 0) && (year % 100 != 0) || (year % 400 == 0)

#define base\_yr 1990

#define base\_mn 1

#define base\_dy 1

int day\_of\_year(int year, int month, int day)

{

switch (month - 1) {

case 11:

day += 30;

case 10:

day += 31;

case 9:

day += 30;

case 8:

day += 31;

case 7:

day += 31;

case 6:

day += 30;

case 5:

day += 31;

case 4:

day += 30;

case 3:

day += 31;

case 2:

day += (leaps) ? 29 : 28;

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

case 1:

day += 31;

case 0:

break;

default:

printf("error!\n");

}

return day;

}

int daytodate(int \*pdays, int year)

{

int month;

int mnth[12] = { 31,28,31,30,31,30,31,31,30,31,30,31 };

if (leaps)

mnth[1] = 29;

int i = 0;

for (; \*pdays > 0; ++i) {

\*pdays -= mnth[i];

}

month = i;

\*pdays += mnth[i-1];

return month;

}

void date\_after\_n(int year, int month, int day, int n)

{

n += day\_of\_year(year, month, day);

while (n > ((leaps) ? 366 : 365)) {

n -= (leaps) ? 366 : 365;

++year;

}

month = daytodate(&n, year);

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

}

int main()

{

int yr, mon, dy;

int n;

while (scanf\_s("%d %d %d %d", &yr, &mon, &dy, &n)) {

date\_after\_n(yr, mon, dy, n);

rewind(stdin);

}

system("pause");

return 0;

}

（5）

//年月输出月历

#include <stdio.h>

#include <stdlib.h>

#define leaps (year % 4 == 0) && (year % 100 != 0) || (year % 400 == 0)

#define base\_yr 1990

#define base\_mn 1

#define base\_dy 1

int day\_of\_year(int year, int month, int day)

{

switch (month - 1) {

case 11:

day += 30;

case 10:

day += 31;

case 9:

day += 30;

case 8:

day += 31;

case 7:

day += 31;

case 6:

day += 30;

case 5:

day += 31;

case 4:

day += 30;

case 3:

day += 31;

case 2:

day += (leaps) ? 29 : 28;

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

case 1:

day += 31;

case 0:

break;

default:

printf("error!\n");

}

return day;

}

int day\_dif\_year(int year, int year2)

{

int days = 0;

for (; year < year2; ++year) {

days += (leaps) ? 366 : 365;

}

return days;

}

int daytowk(int year, int month, int day)

{

int diff = day\_dif\_year(base\_yr, year)

- day\_of\_year(base\_yr, base\_mn, base\_dy)

+ day\_of\_year(year, month, day);

int week[7] = { 1,2,3,4,5,6,7 };

return week[diff % 7];

}

int daytodate(int \*pdays, int year)

{

int month;

int mnth[12] = { 31,28,31,30,31,30,31,31,30,31,30,31 };

if (leaps)

mnth[1] = 29;

int i = 0;

for (; \*pdays > 0; ++i) {

\*pdays -= mnth[i];

}

month = i;

\*pdays += mnth[i-1];

return month;

}

void date\_after\_n(int year, int month, int day, int n)

{

n += day\_of\_year(year, month, day);

while (n > ((leaps) ? 366 : 365)) {

n -= (leaps) ? 366 : 365;

++year;

}

month = daytodate(&n, year);

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

}

int day\_of\_month(int year, int month)

{

int mnth[12] = { 31,28,31,30,31,30,31,31,30,31,30,31 };

if (leaps)

mnth[1] = 29;

return mnth[month - 1];

}

void printmon(int year, int month)

{

printf("%-2d SUN MON TUE WED THU FRI SAT\n", month);

int mondys = day\_of\_month(year, month);

int wk\_of\_fsrt = daytowk(year, month, 1);

switch (wk\_of\_fsrt) {

case 7:

printf("\t");

break;

case 1: case 2: case 3: case 4: case 5: case 6:

for (int i = 0; i <= wk\_of\_fsrt; ++i) {

printf(" ");

}

}

int tab = wk\_of\_fsrt;

for (int i = 1; i <= mondys; ++i, ++tab) {

printf("%3d ", i);

if (i != mondys && tab == 6) {

printf("\n");

printf(" ");

tab = -1;

}

}

for (; tab - 1 <= 7; ++tab) {

printf("\t");

}

printf("\n");

}

int main()

{

int yr, mon;

while (scanf\_s("%d %d", &yr, &mon)) {

printmon(yr, mon);

rewind(stdin);

}

system("pause");

return 0;

}

004 1 
4 
11 
18 
25 
004 12 
5 
12 
19 
26 
1 SUN MON TUE WED THU FRI SAT 
7 
14 
21 
28 
12 SUN MON TUE WED THU FRI SAT 
8 
15 
22 
29 
5 
12 
19 
26 
6 
13 
20 
27 
6 
13 
20 
27 
7 
14 
21 
28 
8 
15 
22 
29 
2 
9 
16 
23 
30 
2 
9 
16 
23 
30 
3 
10 
17 
24 
31 
3 
10 
17 
24 
31 
4 
11 
18 
25 

7.

//输出年历

#include <stdio.h>

#include <stdlib.h>

#define leaps (year % 4 == 0) && (year % 100 != 0) || (year % 400 == 0)

#define base\_yr 1990

#define base\_mn 1

#define base\_dy 1

#define true 1

#define false 0

int day\_of\_year(int year, int month, int day)//日期转换为在这年对应的天数

{

switch (month - 1) {

case 11:

day += 30;

case 10:

day += 31;

case 9:

day += 30;

case 8:

day += 31;

case 7:

day += 31;

case 6:

day += 30;

case 5:

day += 31;

case 4:

day += 30;

case 3:

day += 31;

case 2:

day += (leaps) ? 29 : 28;

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

case 1:

day += 31;

case 0:

break;

default:

printf("error!\n");

}

return day;

}

int day\_dif\_year(int year, int year2)//两年之间的天数差

{

int days = 0;

for (; year < year2; ++year) {

days += (leaps) ? 366 : 365;

}

return days;

}

int daytowk(int year, int month, int day)//日期对应的星期

{

int diff = day\_dif\_year(base\_yr, year)

- day\_of\_year(base\_yr, base\_mn, base\_dy)

+ day\_of\_year(year, month, day);

int week[7] = { 1,2,3,4,5,6,7 };

return week[diff % 7];

}

int day\_of\_month(int year, int month)//一个月有几天

{

int mnth[12] = { 31,28,31,30,31,30,31,31,30,31,30,31 };

if (leaps)

mnth[1] = 29;

return mnth[month - 1];

}

void prnt(int wk\_of\_fsrt)//打印一行的月份中1号之前的空白

{

switch (wk\_of\_fsrt) {

case 7:

printf(" ");

break;

case 1: case 2: case 3: case 4: case 5: case 6:

for (int i = 0; i <= wk\_of\_fsrt; ++i) {

printf(" ");

}

}

}

void printline(int ln, int year)//输出一行的两个月份

{

int lmonth = ln;

int rmonth = ln + 6;

printf("|%-2d SUN MON TUE WED THU FRI SAT ", lmonth);//第一行

printf("%2d SUN MON TUE WED THU FRI SAT |\n", rmonth);

int lmondys = day\_of\_month(year, lmonth);

int lwk\_of\_fsrt = daytowk(year, lmonth, 1);

int rmondys = day\_of\_month(year, rmonth);

int rwk\_of\_fsrt = daytowk(year, rmonth, 1);

int ctl = 1; int ctr = 1;//对应月份天数的控制量

int lenough = false; int renough = false;//对应循环的控制量

int teml = false; int temr = false;//确保左右月份已经打印完

while (!lenough || !renough) {

printf("|");

int ltab = 0;

if (ctl == 1) {

prnt(lwk\_of\_fsrt);

ltab = lwk\_of\_fsrt % 7;

}

else if (ctl != 1)

{

printf(" ");

}

for (; ctl <= lmondys; ++ctl, ++ltab) {

printf("%3d ", ctl);

if (ctl == lmondys) {

lenough = true;

teml = true;

++ctl;

++ltab;

break;

}

if (ctl != lmondys && ltab == 6) {

//++ltab;

++ctl;

break;

}

}

if (lenough && !renough) {

if (teml) {//在左边天数已经全部打印出来的且右边还没打印完的情况下

for (; ltab <= 6; ++ltab)

printf(" ");

}

}

if (!lenough && renough) {

for (; ltab <= 6; ++ltab)

printf(" ");

}

if (lenough && renough) {

for (; ltab <= 6; ++ltab)

printf(" ");

}

printf(" ");

int rtab = 0;

if (ctr == 1) {

prnt(rwk\_of\_fsrt);

rtab = rwk\_of\_fsrt % 7;

}

else if (ctr != 1)

{

printf(" ");

}

for (; ctr <= rmondys; ++ctr, ++rtab) {

printf("%3d ", ctr);

if (ctr == rmondys) {

renough = true;

temr = true;

++ctr;

++rtab;

break;

}

if (ctr != rmondys && rtab == 6) {

//++rtab;

++ctr;

break;

}

}

if (lenough && !renough) {

if (temr) {

for (; rtab <= 6; ++rtab)

printf(" ");

}

}

if (!lenough && renough) {

for (; rtab <= 6; ++rtab)

printf(" ");

}

if (lenough && renough) {

for (; rtab <= 6; ++rtab)

printf(" ");

}

printf("|");

printf("\n");

}

}

void printyear(int year)

{

//printline(4, year);

printf("|------------------The Calendar of Year %4d ---------------------|\n",year);

for (int line = 1; line <= 6; ++line) {

printline(line, year);

}

}

int main()

{

int yr;

printf("Please input the year whose calendar you want to know: ");

while (scanf\_s("%d", &yr)) {

printyear(yr);

rewind(stdin);

printf("Please input the year whose calendar you want to know: ");

}

system("pause");

return 0;

}

计算机生成了可选文字:
Please input 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
SUN MON TUE WED THU FRI 
the 
6 
13 
20 
27 
3 
10 
17 
24 
2 
9 
16 
23 
30 
6 
13 
20 
27 
4 
11 
18 
25 
8 
15 
22 
29 
the 
year whose calendar you want 
The Calendar of Year 2004 
to know: 
2 
3 
4 
5 
6 
4 
11 
18 
25 
8 
15 
22 
29 
7 
14 
21 
28 
4 
11 
18 
25 
2 
9 
16 
23 
30 
6 
13 
20 
27 
5 
12 
19 
26 
2 
9 
16 
23 
8 
15 
22 
29 
5 
12 
19 
26 
3 
10 
17 
24 
31 
7 
14 
21 
28 
7 
14 
21 
28 
4 
11 
18 
25 
3 
10 
17 
24 
31 
7 
14 
21 
28 
5 
12 
19 
26 
2 
9 
16 
23 
30 
year 
8 
15 
22 
29 
5 
12 
19 
26 
4 
11 
18 
25 
8 
15 
22 
29 
6 
13 
20 
27 
3 
10 
17 
24 
2 
9 
16 
23 
30 
6 
13 
20 
27 
5 
12 
19 
26 
2 
9 
16 
23 
30 
7 
14 
21 
28 
4 
11 
18 
25 
SAT 
3 
10 
17 
31 
SAT 
7 
14 
21 
28 
SAT 
6 
13 
20 
27 
SAT 
3 
10 
17 
SAT 
8 
15 
22 
29 
SAT 
5 
12 
19 
26 
7 
8 
9 
10 
11 
12 
4 
11 
18 
25 
8 
15 
22 
29 
5 
12 
19 
26 
3 
10 
17 
24 
31 
7 
14 
21 
28 
5 
12 
19 
26 
5 
12 
19 
26 
2 
9 
16 
23 
30 
6 
13 
20 
27 
4 
11 
18 
25 
8 
15 
22 
29 
6 
13 
20 
27 
6 
13 
20 
27 
3 
10 
17 
24 
31 
7 
14 
21 
28 
5 
12 
19 
26 
2 
9 
16 
23 
30 
7 
14 
21 
28 
7 
14 
21 
28 
4 
11 
18 
25 
8 
15 
22 
29 
6 
13 
20 
27 
3 
10 
17 
24 
8 
15 
22 
29 
2004 
8 
15 
22 
29 
5 
12 
19 
26 
2 
9 
16 
23 
30 
7 
14 
21 
28 
4 
11 
18 
25 
2 
9 
16 
23 
30 
2020 
2 
9 
16 
23 
30 
6 
13 
20 
27 
3 
10 
17 
24 
8 
15 
22 
29 
5 
12 
19 
26 
3 
10 
17 
24 
31 
SAT 
3 
10 
17 
31 
SAT 
7 
14 
21 
28 
SAT 
4 
11 
18 
25 
SAT 
2 
9 
16 
23 
30 
SAT 
6 
13 
20 
27 
SAT 
4 
11 
18 
25 
Please input 
whose calendar 
you want 
to know: 