1. **日期相关的问题**

日期相关的问题基本上都可以通过下面的类来完成。

class Date

{

private:

int y, m, d;

int isleap(int year)

{

if (year % 100 == 0)

return year % 400 == 0;

else

return year % 4 == 0;

}

public:

Date(int \_y = 1900, int \_m = 1, int \_d = 1)

{

y = \_y, m = \_m, d = \_d;

}

void operator ++(int)

{

static int days[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}

};

d++;

if (d > days[isleap(y)][m - 1])

m++, d = 1;

if (m > 12)

y++, m = 1;

}

bool operator == (Date& date)

{

return y == date.y && m == date.m && d == date.d;

}

void print()

{

cout << y << "-" << m << "-" << d << endl;

}

};

**比如求2014-11-9后1000天。**



Date d(2014,11,9);

for (int i = 0; i < 1000; i++)

d++;

d.print();

1. **满足条件的数字**

**分解一个整数的各个数位，存放在一个数组中。**

int num = 1234567;

int digit[10];

int i = 0;

while (num)

digit[i++] = num % 10, num /= 10;

**判断一个数字的各个数位数字均不相同。**

int num = 1234567;

int times[10];

memset(times, 0, sizeof(times));

while (num)

times[num % 10]++ , num /= 10;

bool ok = true;

for (int i = 0; i < 10; i++)

if (times[i] > 1)

ok = false;

times记录了这个数字中每个数位上数字出现的次数，各个数位数字均不相同即是不存在任何一个数字出现的次数大于1。很多判断一个数字是否满足某一条件的问题都要用到这种方法(用空间换取时间)。

1. **排列生成的方法**

**生成1，2，3，4，5的所有排列**

int arr[] = {1, 2, 3, 4, 5};

do

{

//处理代码

}while (next\_permutation(arr, arr + 5));

这段代码几乎每次比赛都会使用！