## 素数（质数）：

在所有比1大的整数中,除了1和它本身以外,不再有别的约数

2、任何数%100都小于100

任何数%N都小于N（求一个数的个位%10）

3、四位数除100，得两位数

比如int中 1950除100，得19

19%10 = 9

4、/\*不同进制的10或其他相同数字是不一样的\*/

要解决 X进制 + X进制 的进位问题，都可以

先把X进制转化为10进制，把两个10进制数相加的和

再转化成X进制，就可以解决进位问题

5、如果字符a的值为'0'到'9'这10个字符之一，则a-48相当于将字符转换为数值。

若a='5',则a-48=5。其中48是'0'的ASCII码。

6、1不是素数（质数），所以素数（质数）是从2开始的。

7、当前数可以由前一个数算出来的情况，可以用数组，初始化最前面的元素，后面的数根据前面的数进行规律运算计算出来。

## 8、精度相关问题

（1）整数除整数，默认就会得到一个整数。如 100 / 53 这个整体，就会默认是1。

（2）浮点数除整数，向会把整数变成浮点数，如 (double)100 / 53

或者100/（double）53，这个整体会获得一个浮点数。

（3）double和float类型默认精确到6位。

## 9、全排列与深度优先遍历：

（1）全排列实现一般不涉及到标志数组，模型如下：

void printfAllSort(int k) {

if (k == n)

{

cout << str << endl;

return;

}

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

{

swrap(i, k);

printfAllSort(k + 1);

swrap(i, k);

}

}

10、#include <stdlib.h>包含的函数：

(1)abs(long i);

## 10、输入跳过问题：

可能存在输入时，跳过输入的情况：

解决方法:

1. 尝试把cin替换成scanf

## 11、等差数列求和公式：

* formula

## 12、逆序对：

如果存在正整数 i, j 使得 1 ≤ i < j ≤ n 而且 A[i] > A[j]，则 <A[i], A[j]> 这个[有序对](http://baike.baidu.com/view/6055238.htm)称为 A 的一个逆序对，也称作逆序数

## 13、黄金分割数和斐波那契数列的关系：

（1）菲波那契数列与黄金分割有什么关系呢?经研究发现,相邻两个菲波那契数的比值是随序号的增加而逐渐趋于黄金分割比的.即f(n-1)/f(n)-→0.618….由于菲波那契数都是整数,两个整数相除之商是有理数,所以只是逐渐逼近黄金分割比这个无理数.但是当我们继续计算出后面更大的菲波那契数时,就会发现相邻两数之比确实是非常接近黄金分割比的.

（2）因此，可以利用斐波那契数列求黄金分割比。

（3）斐波那契数列一般使用迭代求法，速度快很多

（4） // 模拟除法过程

int ans;

for(int i=0; i<101; i++) {

ans = x / y;

cout<<ans;

x = (x%y)\*10;

}

注意，第一个算出来的不是小数位，是个位。