1. 0 并不是什么都没有 0 是占位符

有些语言把没初始变量设为0 把没初始化的对象设为null

布尔中 一般是true1 false0 if()语句会把返回值转换为布尔类型，大于0的转换为1，小于0 的转为0

数字转字符串， 字符数组转字符串

Char\* 可以指向字符串 char\* =”” char[]=”abc” strlen(char) 长度不包括\a char\*是常量区不可修改（指向字符串常量）

字符数组没有结束标志，所以不能加上字符串 strlen靠读取结尾获取长度

数组名保存了什么信息 sizeof(a) sizeof(int) 类型长度

有的语言不初始化就报错

有的语言直接给变量分配内存，不初始化的话显示的是内存中原来的值（局部变量内存是栈形式，不清除的）

存在类型转换的隐式运算符 比较 赋值 加减乘除 字符串连接 会转成字节长的 char和short运算会转整型 所有浮点运算都是双精度，会先转换再转回

赋值会丢失精度，此时会警告，应该进行强制类型转换就不会报了

关于类型转换

把字符串设为”” 即 00 字符串结束符\0

3、除0异常 异常捕获处理 显示错误，不中断程序

Try catch

4、十进制 乘以十，左移 除以十左移 移位法 取得位上的数字mod10

二进制 乘2 移位 除2 获得最后一位 模2

一级运算 二级运算 先乘除后加减 加法：位与位加，逢十进一 二进制是逢二进一

如果有进位，把1和下一位两数加，a+b+1 如果结果还是大于十，再次进位

二进制一样 大于2就进一，下一位是三个数相加a+b+1

减法是加法的逆运算，如果 结果位相减小于零，则从下一位借位,多减一个1。。。a-b-1

二进制加法只有1+1或者1+1+1 会进位 结果是0进1 或1进1

只有0-1或者0-1-1会借位，借一当二，当前位当成2-1=1

如果下一位仍然是 0-1 本位就是2-1-1=0 下位继续借 a-b-1

另外减一个数就是加上负数。。。。这样可以把减法转化为加法，关键是负数怎么表示 可以用反码法，反码是最大值减当前值 由于最大值任何一位都大于其他数，不需要借位，减法很容易

199-256 = 199 +（-256） =199+999+-256 +-999 =199+743 -999

取反，本质上是最大值减掉当前值 99999-000123 在二进制中直接反位就行了，为什么？ 反码=111111111 - xxxxxxxxx 可以发现 1-1=0 1-0=1

乘法是逐位乘12\*23 =12\*2\*10+ 12\*3 乘法表原理，把乘数和被除数每一位相乘 然后把所有结果**移位后**相加

转化成多位数乘一位 的和 多位数乘一位的原理：各位和一位数相乘，然后**移位**相加

转化成个位数乘个位数 个位数相乘原理，N个M相加 （用99乘法表化简），每次相加都看是否进位（和大于10），大于十则进位+1

乘法其实是多次加法，12+12+12循环23次 不过这样并不聪明 使用右移？

二进制乘法：1\*0 =0 1\*1=1 只有一种情况会等于1，而且**单位相乘没有进位**

所以可以简单地 拆成 乘数分别乘 被乘数每位然后 移位相加

被乘数每位只能是0或1 所以很简单，被乘位是0，则置0 被乘位为1，则保留乘数 把乘数 多次左移， 根据被乘位为0还是1判断+0 还是加上被乘数

For () a<<1 if(b[n]==0) x+0 if(b[n]==1) x+=a

问题来了，如何根据1和0来决定是保留0还是保留原数呢？

Xor 异或 同为 0 异为1 ：和0异或 和0 异或则结果为原数 和1 异或似乎只改一位？ 和自己异或则得0

除法由**因数积**求因数 是乘法的逆运算 实际上是减法

循环连续减去被除数，如果剩下的比被除数大，则返回 如果剩下的数比被除数小，则

但是减是从最高位减的，原理呢？ 乘法是多乘1的结果a,b,c 移位相加 a\*100+b\*10+c\*1

相加后根据abc的值产生进位

，除法自然是 要拆分 先对前面的拆分 d\*100+e\*10+f xxxxx00000

D 是a +进位值 a能被除数整除，而进位不能被整除，是余数

我们 对d做除法（连续减法），获得一个商(等于a/除数 )和余数 余数和

E相加 ，继续除商

除法运算： 0/0 1/0 不存在 无意义

取余，模的周期循环…… 一个**模是指被除数 ，其长度就是数字循环的规律，任何数字针对一个模数取模，都会呈现出周期性** 任意两个同余的数

十进制分解 1+ 10+10^2+10^3

**二进制：每一位都代表2^n 1，0表示2^n的个数，1的个数，2的个数，8的个数**

**奇数：末位为1 偶数 末位为0**

**左移是把各位的指数加1 最后2^(n+1)+2^n……= 2^n \*2 +2^(n-1)……**

**即整个数字\*2 右移是/2** 如果减去某一位的1就代表删除2^n 如果加上某位的1 就是增加2^n 用1减去某位代表对某位取反，即如果是0则加上2^n 如果是1则减去2^n 所以1111111111-xxxxxx是对所有位取反

**如果某位不确定是否为0，加某位的一和上面仍然是一样的，是整个数+2^n 可能会存在进位 如果进位时遇到一串一，就会把所有一都变成0，直到 遇到0，把0变1**

1111111 +1 =1000000 同样的

减去某位的1 是一样的， 是整个数-2^n 如果发生借位 0-1的情况 前面遇到一串00000则会一直到第一个1 。把1变为0

二进制位：1位 2位 4位 8位 16位 32位 位名表示 1000000时的值，即该位值

比如2^5 是32位，10000表示32

00000~01111 表示 0~31 最前面的1是一个位的分界标记，当最高位为0 后面的位表示 0~31 当最高位为1，则总数表示 32+ 0~31 以 1位 作为分界阅读是很合理的……

一个n位数表示的最大值 1+2+4+8+……2^n =2^(n+1) -1 11111就是2^6-1=63

数长度限制 32位中long和int一样4字 64位中long 8字节

大整数实现：利用数组，多字节 利用字符串 实现add sub mul div mod等方法

除法需要试商 加法？取位 移位

电脑用二进制，只用判断电平高低（有无），不用判断大小

八进制 十六进制 用一位 表示三到四位的二进制数 o hex

常用于缩短表示二进制数 方便阅读，

八个二进制位是一个字节，两个十六进制位就可以表示一个字节 从00到ff 表示0~256

十二进制：时钟 绕一圈又从零开始

位的使用方式： 不同数字区分不同状态 0~255就是255个状态值比如00~ff表示256种颜色深度

使用多位组合表示复合状态 ，使用位表示每种单独的状态 前提是每种状态只有两个值

用数字映射概念 或者是第一位表示黑白，第二位表示高矮 ，第三位表示大小

小高黑 大矮白

很多时候一位两种状态是不够的，我们通过多位表示状态的多个值 然后把状态按类分组 比如rgb 红ff 绿 ff 蓝ff 分别都是256个颜色深度

1. 素数 不能作为任何数因子的自然数 1和0不是素数也不是合数

1、素数判断 2 3 5 7 11 素数并不是很多

确定性算法很慢 随机算法用计算机解决

试除法： 除以从1~N-1的所有 数，看有没有余数 ，发现—— 无需验证<根号n的值 无需验证合数

所有合数都是素数组成的 所以合数最终一定能拆分成几个素数的积，可能是两个，或者更多

Is\_prime(n) n<0 0 1时返回错误信息

2、筛子法 （对于查找有限个的素数有用） 筛掉找到的所有的素数的倍数

需要比较大的空间保存所有的数

**用数组下标表示每个数 下标中保存状态,非素数为0 素数为1**

3、用 多一个数组保存所有素数表

定理 (a,2a)间存在素数 存在等差数列素数（长度不定） （到一定长度就会出现合数）

一个偶数写成两个数字和，最多9个质因 = 一个质+一个合 最多五个因子

一个大偶 素数+最多两个质因子

孪生素数：相差2 q和q+2 不能被< sqrt(q+2)整除

4、中国剩余定理 韩信点兵

%3=2 %5=3 %7=2

列出所有满足条件的数，求公共值

条件合成 ——公共值必然是周期性的 因为余数是周期的，需要把两数的周期合并

如果除数互质则取积为周期 如果除数是因子则取最大除数

周期的起点是第一个满足该状况的数，也就是新的余数要求

%15=8

然后和第三个数合成

%105 =23

4、找出~1000的所有孪生素数 ：从素数中筛选

5、RSA算法 （使用素数）

对称加密 用一个key通过某种变换得到密文 简单的就是移位，异或等

（被变化的可能是一位，也可能是多个字节，**注意下位变化的本质是什么数学变化**）

非对称 公钥publish key 密钥 s key 公钥加，私钥解

Rsa定理： p q 互质数 r m正整 m与(p-1)(q-1)为质 （公因数为1 ）

当Rm % (p-1)(q-1) ==1

有正整数A<PQ C=A^r % pq B=C^m %pq

此时A=B （A^r % pq）^m %pq ==A

共有四个量 p q r m 可以把一个数变为另一个 需要分别得(p-1)(q-1)和pq的值

A作为明文 C作为密文 pq是某种固定量 ,r，m就是key了 r是公钥，m是密钥

Pq随便选。 选择一个密钥m (<(p-1)(q-1)互质) ，由公式生成公钥r ——发布公钥r 和 pq （作为整体）

（次方往往很大。需要使用大整数运算（数组或者字符串）

数字签名技术：私钥运算，公钥验证

数据分组加密（**只加密其中几位，或者是几字节——最好是不要让字节断开**），最后拼合起来 安全性和pq的位数有关 1024位密钥 最好使用2048位的pq（256位的16进制）

2.6 歌德巴赫猜想

充分大的偶数，可以表示成素因子不超过 a b个的 两数和 a+b

陈：可以表示为两个素数和，或者一个素数和半素数和

两个质数和

素数是不规律的

验证： N=N1+N2（都是正整数） n1 n-n1 从2开始 一直到n/2

2.5，梅森素数

2^p -1 P为素 2^p -1也为素 M11不是 只发现48个

1. 递归 自己调用自己

从前有座山，山里有个庙…………

1. 德罗斯特效应 画里面还有一副画 电视里还是电视

2、递归 **同类对象 的嵌套 和重复 递归对象间具有层级关系**

**递归对象间的 有着一种递进的关系 ……比如指针引用，索引前进，参数增大**

1. 怎样移动100个箱子 移动箱子（100） {移动一个箱子（）; 移动箱子（99）}

数学中有这种重复的推演 ：自然数的后继

3、何时递归 一般递归的问题需要 **问题本身就是递归的层次 如：二叉树，文件目录 递归定义**  递归问题需要能被划分为子问题 （集合——子集合） 如二分查找，划分为子数组

自然界很多递归问题没有尽头 ，但是程序中必须有

（1）每次递归都应该缩小问题规模 （2） 前后递归之间有某种关系， 比如前次的输出作为后次的输入 a(i){a(i+1) } a(p) { a(p->next) } （3）每次递归都要设置结束条件 **需要分解问题，需要边界**

1. 最大公约数

A能被b整除 b 是约数 几个数公约数中的最大值称为最大公约数

（1）辗转相除法 欧几里得 大数除小数 m%n =r n%r=r2 r%r2=

一直到出现整除

每次都是规模缩小，操作相同，终止条件为 余数0

循环写法和递归写法的区别： 一个是while(){ m=n n=r 如果余数0 return } 递归：return 0 return gcd()

4、递归的return return后面是一个数值 最终值会**一层层地向上传递**为最下层的值 如何用其他的方式实现？使用全局变量修改 (1) return a() 最终值是最后的a （2）return a()+n 最终值是最后的值不断地变化 （3）多个方向的return if() return a() else return a() （4） 循环+递归 反复执行 多次递归 a(){ for{ a() } } 循环中应该是有控制变量的 可以改变内部递归的路线 **循环本身常用在遍历，配合递归实现深层遍历，循环递归，递归中又循环……比如打印文件夹下各层的文件**

**递归是返回值 还是停止 break 是set 还是get**

**Set可以修改指针类型实现修改非全局变量**

**递归：修改参数 修改调用路径 a.next() b.next()**

**调用不是自己的函数，但是函数中 存在间接调用自己的可能，同样是递归**

5、阶乘

《程序员的数学》

一、0是占位符

1. 逻辑 为了消除歧义 什么是歧义？

陈述句 表示命题 a>0 命题具有真假 a>0 ==true

真假由规则进行判断 被分析的**属性a 条件>10 if (a > 10) then**

对于分析的问题

是否存在遗漏？ 有考虑边界吗？ 大于小于 第一 最后

If (10>a>0>b>-1) 对于等于0的情况 等于1 -1的情况没分析

命题是否有重叠部分 条件是否重复

，重复部分是否矛盾？（矛盾是指满足条件时执行的动作有矛盾，对同一属性产生了不同的操作，如果是操作不同的属性，那就没矛盾）

If(a>10) a= 1 if（a>6） a= -1 当a在6到10之间会有矛盾

重叠的范围是哪些？

编程语言中，条件语句顺序执行，所以矛盾的语句会执行后面的

可以用横坐标轴分析是否存在遗漏和矛盾

**规则条件没有遗漏，就可以适应属性的任何情况，在软件测试中表示覆盖率**

对所有的情况都进行应该有的处理，比如输入小数，输入字符时，输入负值，输入0，正确输入和错误输入都应该得到正确处理，（当然，本来就不需要处理的状况就可以省略，但是要知道仍然有得到错误输入的可能性）

一般来说，输入错误类型是会转型的，但是错误的输入本身就不会得到正确的结果，这部分我们看情况是否能够处理，给出错误提示

3、分解问题，根据命题的真假分成两份 if(a>5) if(a<=5)

如果有需要还可以继续分解 if( 3<a<=5) if(a<=3)

4、命题的条件可能是不连续的 if a==1 if a==2 switch

5、复杂命题 具有多个属性的命题 if(a>1 && b>2) 是多个小命题的多个属性组成的 小命题之间也存在逻辑关系

6、文氏图 小命题属于**属性的集合**  其中 有n个元素属于true m个元素属于false（nm可能是无限，代表了所有情况的集合如代表所有a>0=true ） 现在就是求集合true false部分的交集

集合之间的关系，交并补

圆圈的交集 并集 全集 阴影法 相等 异或

二灯游戏 a绿灭黄亮 b都灭 c都亮 状态a=1/0 b=1/0

(-a)&&b -a&&-b A&&b

卡诺图：表示所有命题组合

7、第三个逻辑 未定义 undefined

带条件与Ab都不是undefined时和原来一样

如果a是false B是undefined 发生短路 最后是false

其它情况AB任何一个是undefined返回undefined

带条件与表示后续是否执行 遇到false和undefined都会导致阻断

或在遇到false时不阻断，遇到undefined时阻断，最后结果是undefined

但是 undefined在尾部时 true>undefined>false

Undefined否定 :undefined

1. 除法是分组

奇数，偶数 /2

/10是把数分成十组 **每组数都是等差的 除数称为模**