Factorization: 积分，把一个自然数分解成一串质数prime的乘积

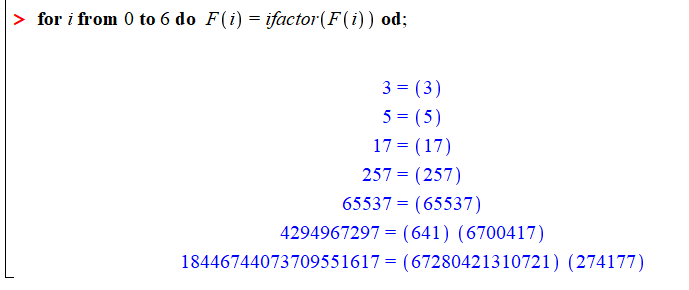
Fundamental Theorem of Arithmetic:任何自然数都可以分解成一系列prime的乘积，且这个factorization是唯一的

使用Ifactor

有没有方法计算所有prime number?

方法1：Fermat Numbers:

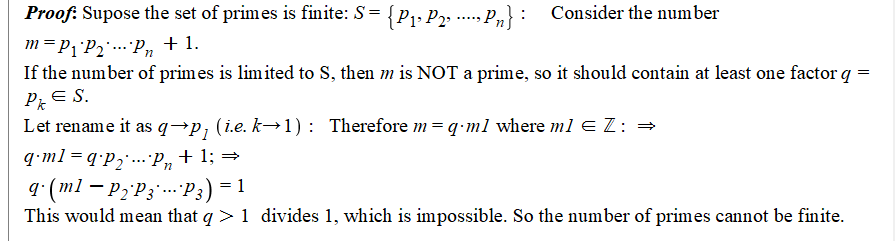




算到第五阶就不行了

Euclid's theorem：有无数prime

证明：就是那个+1法



Primes in an interval

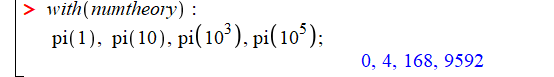
在一个区间内有多少个质数

定义



pi(x)代表着小雨x的指数数量

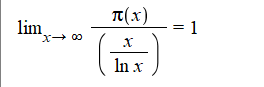
maple怎么用



激活数论，然后直接pi(x)

理论1

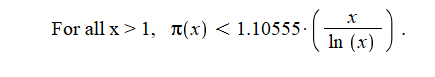
Hadamant & de la Vallee poussin



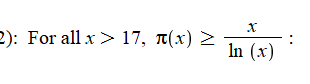
当x无穷大时

质数的数量大概与x/lnx差不多

理论2：



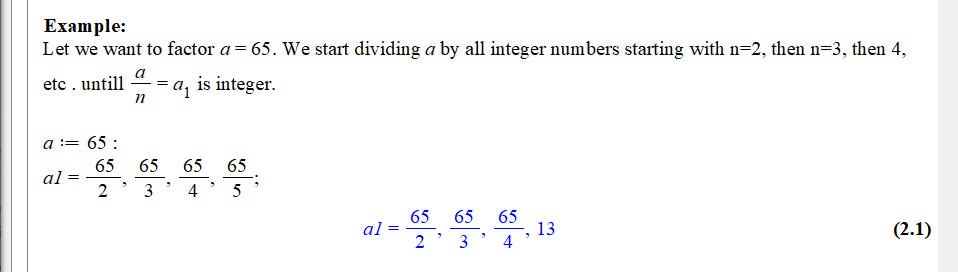
理论3：



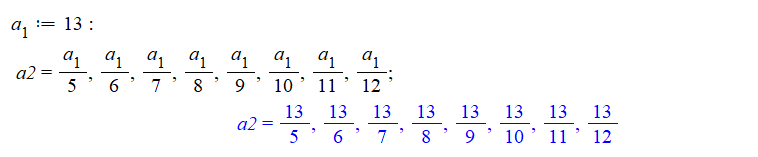
Trial Division , Sieve of Eratosthenes

一个简单的factorization的想法是Traill division

从最小的质数开始尝试约分，一直到下一个最小质数被找到，每一次尝试叫trial division， 当找到一个成功整数后，下一轮trial从a1=a/p1, 最小质数p1开始，重复这一过程直到约分完毕



在4个trial后我们找到p1=5,



然后从13/5开始

最后另外一个factor为13

因此就是65=5\*13

所以我们一共用了12个trial divisions<

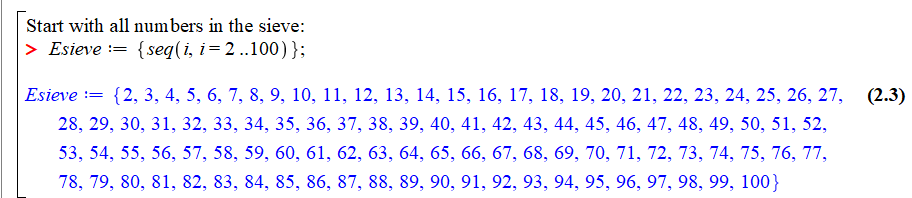
这个过程可以被缩短如果我们只用质数来测试

但是我们需要知道

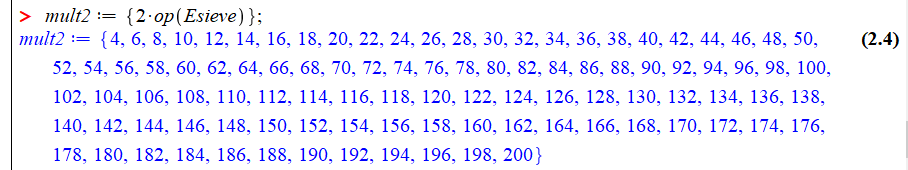
一个简单地找到[1,n]之间所有质数的方法叫做sieve of Eratothenes

Sieve of Eratothenes，：

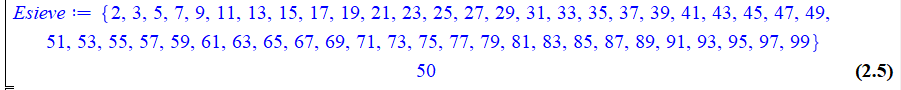
step1:列出一个集合，聚集了从2到n的所有数

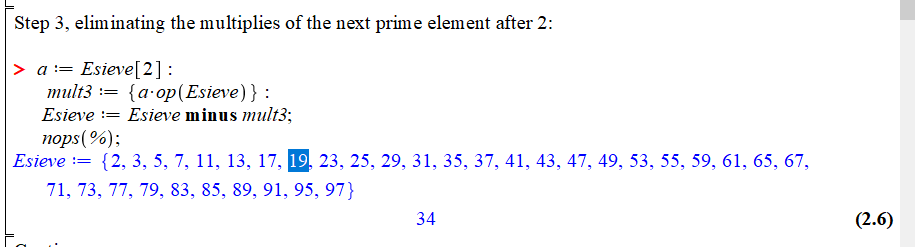


step2:让这个集合乘以2



Step3:让两个式子相减

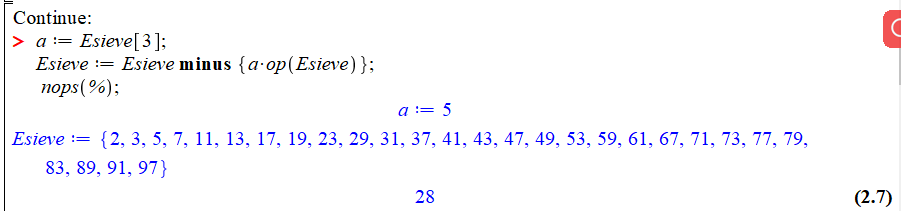
//成功排出2的倍数



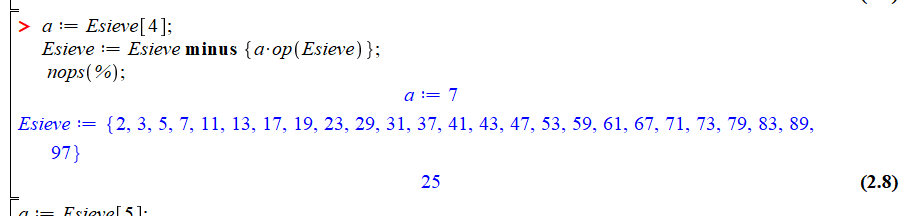
然后a要取Esieve第二个数

op操作符，把集合里的所有数依次提取出来//注意这里的Esieve，是STEP3之后的

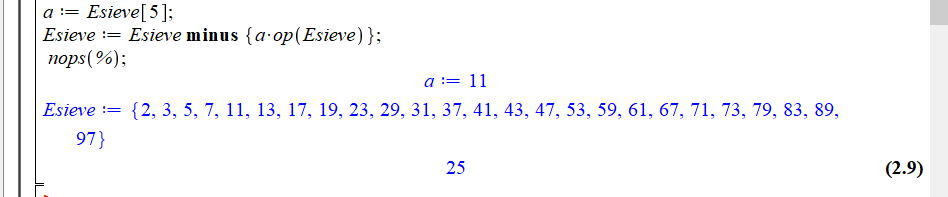
NOPs(％)，看有几个element



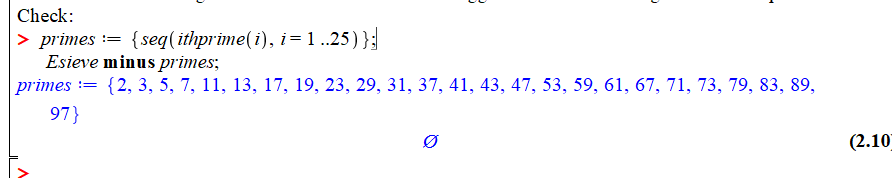
然后取5的倍数



就循环就行



测试是不是都是prime

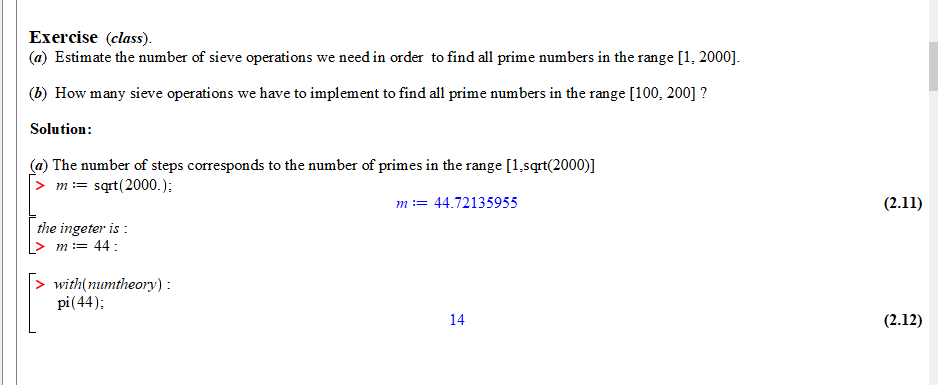


**这里多测了一步**

实际上只要测完Esieve[4]就行

我们只要测完<=根号n 的所有质数，就行

例题



首先我们要测的质数必然小于等于最大数根号=44，

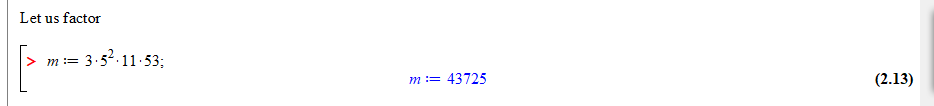
44里的质数用pi(44)得到14各直属

所以我们需要14步得到所有prime

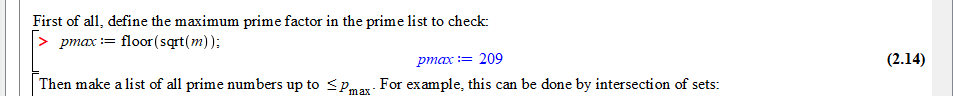
Factorization by Trial division:

为了通过trial division进行factorization，我们需要预先知道从2到根号m之间的所有prime 数，然后我们切割m一直到找到一个factor，然后从ma=m/p1继续 trial division

假设m=43725=3\*5^2\*11\*53

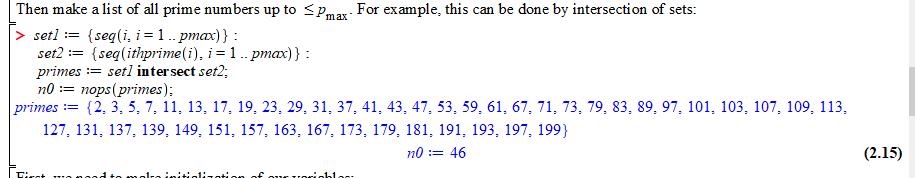


Step1:找到封顶数



sqrt m 取整

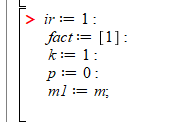
Step2,找到这之间的prime number



可以用seq(ithprime(i),i=1…pmax直接求出Prime

Intersect,交集，

然后初始化变量:



fact:factor的意思 ，所有prime因数的集合

k:第几个质数

m1：当前被除数

p: 质数具体值

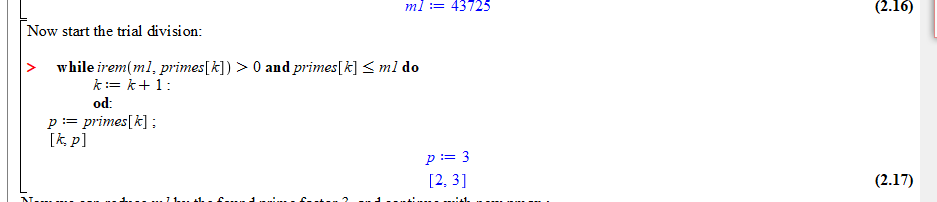
maple指令解释

while loop,

Irem(23,4) 得到余数 3

iquo（23,4）得到商 5

od是 end do 的意思

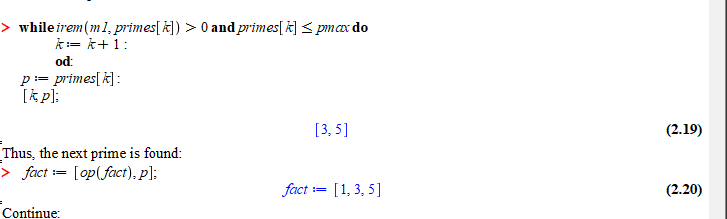


因此直到找到下一个质数为止，一直让k++

然后p的具体值

然后factor加上p，

m1=m1/p

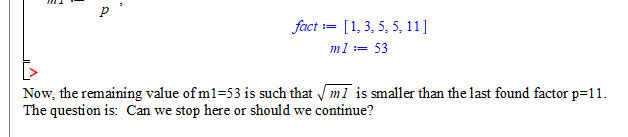


然后不停循环

注意如果成功既m1可以整出prime[k]不会k++， 这样就能满足重复的k的情况

什么时候停止：

根号m1小于最新找到的factor

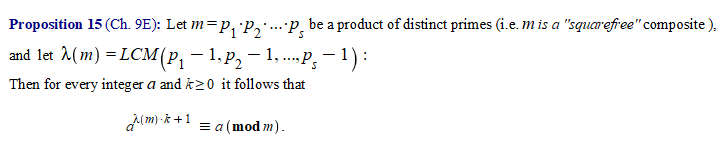


RSA cryptography

Euler Theorem非常依赖RSA

记得：

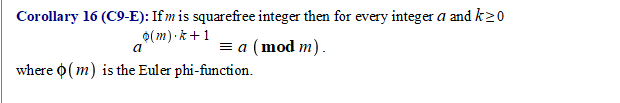




如果m是一串不重复的质数的乘积，那么入（m）=LCM(p1-1,p2-1,..ps-1); 最小公倍数

那么a^(入(m).k+1)≡a(mod m)

推论16：

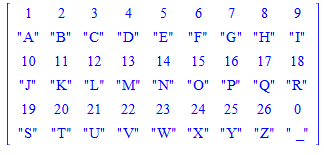


如果m是一个squarefree integer， 那么对于任何a与K,



RSA encoding

总而言之就是1到26覆盖字母表，65啥的只是ascii编码



例如A就是Char(65)的结果

选取阶段：

假设mary想发信息，而Peter想接受，那么peter就需要选两个比较大的质数prime， 且让mod数m=p\*q, //PQ 长度要一样， 那么 phi (m)=(p-1)(q-1) //公式求出来的。然后Perter还需要选一个与phi（m）互质的数字e，这样e就是phi(m)为mod数时 的Unit

这样ed≡1(modphi(m))是有解的



选好后把条件发给mary

对于某个数k， Peter保存dpq让他们secret，然后把me发给mary

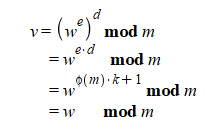
Mary得到所有条件，发送信息阶段

Mary持有信息w(数字形式)，如果这个数字大于m（数字2222 对2111，也要切，不仅是几位），需要切割成block，每一个长度小于m, 我们这里先假设w只有一个block，为了加密w， mary需要使用

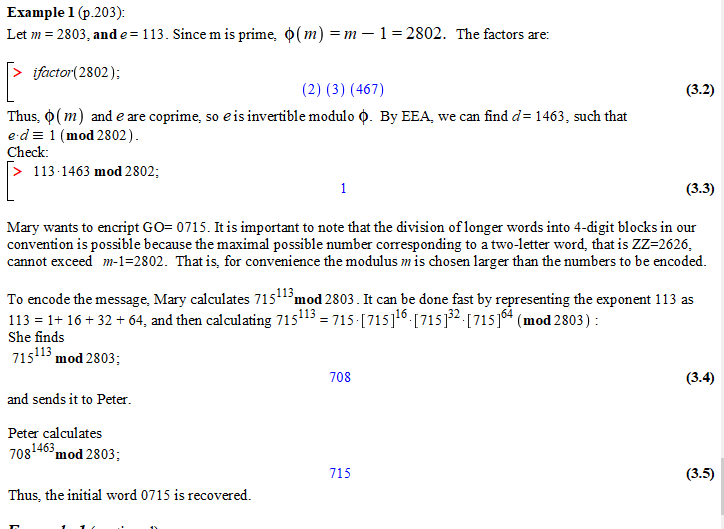
**c=w^e mod m ,并且把c 发给peter**



peter解密使用



例子：



假设m=2803, e=113， 因为m是质数所以phim =2802

因式分解Phim， ，直到与e互质，

因此我们能找到d=1463



假设Mary想发送的信息为0715

小于m-1，不用切

715^113 MOD 2803 加密

=708

peter计算708^1463 mod 2803 =715， 0715 OK