TEORIA DE LOS NUMEROS.

Introducción.

La teoría de los números es, dentro de las matemáticas, la encargada de estudiar las propiedades de los enteros, es decir, la paridad, aditividad, primalidad, multiplicidad y divisibilidad. En este apartado nos dedicaremos a estudiar algunas de las propiedades y teoremas más importantes de este campo, que nos serán de mucha utilidad en el campo de la criptografía.

Definiciones básicas.

A continuación presentamos algunas definiciones y resultados matemáticos que nos serán de utilidad.

Paridad.

Se define la paridad como la propiedad de que un número sea divisible por dos, en cuyo caso decimos que tiene paridad par o que es un *número par*, o no, en cuyo caso decimos generalmente que se trata de un *número impar*. Si en lugar de utilizar la representación decimal utilizamos la binaria es muy fácil decidir si un número es par o impar, solo tenemos que fijarnos en el bit de la derecha, si el bit es un uno el número es impar, en caso contrario se trata de un número par.

Las propiedades más importantes relativas a la paridad son:

Propiedad. La suma de dos números es par si ambos tienen la misma paridad. Generalizando, si sumamos n números enteros, si n es par, el resultado será par si todos los números tienen la misma paridad. Si n es impar, el resultado será par si todos los números son pares e impar si son impares.

Propiedad. La diferencia de dos números es par si ambos tienen la misma paridad. Generalizando, si restamos n números enteros, si n es par, el resultado será par si todos los números tienen la misma paridad. Si n es impar, el resultado será par si todos los números son pares e impar si son impares.

Propiedad. El producto de dos números es par si uno de ellos es par. Generalizando, el producto de n números es par su al menos uno de ellos es par.

Primalidad.

Se define un *número primo* como todo aquel número entero que cumple que solo es divisible por él mismo y por la unidad. Inversamente definimos un *número compuesto* como todo número entero n mayor que 1 tal que $n = a.b$ con $a > 1$ y $b > 1$. A los números a y b según la definición anterior se les denomina *divisores propios*, con lo que podemos definir de forma equivalente un número compuesto como aquel que tiene divisores propios.

Determinar si un número es primo o compuesto no es difícil, simplemente hay que ir dividiendo por los números positivos (asumiremos siempre que el número es positivo) inferiores a él, o mejor hasta su raíz cuadrada (si es compuesto uno de los elementos será mayor o igual y el otro menor o igual), excluyendo el cero y el uno.

El siguiente ejemplo hecho en freebasic puede ayudaros a ver el proceso:

```
' Programa para determinar si un número es primo o no.
Dim As LongInt numero, divisor, limite, Resto
Dim As Double numerof
Dim As Integer I, salto(4)
Cls()
Input "Entra el numero:", numero
'Solo es necesario mirar hasta la raiz cuadrada ya que si es compuesto uno de los números estará por encima y otro por debajo
numerof = numero * 1.0
numerof=Sqr(numerof)
limite=numerof
' comprobamos si es divisible por 2
Resto=numero Mod 2
If numero > 2 And Resto = 0 Then
        Print "Numero divisible por 2"
        sleep
        End
End If
Resto=numero Mod 5
If numero > 5 And Resto = 0 Then
        Print "Numero divisible por 5"
        sleep
        End
End If
' No hace falta que miremos los pares ni los multiplos de 5
salto(1)=2
salto(2)=4
salto(3)=2
salto(4)=2
I=0
divisor=1
While divisor <= limite
    If I<4 Then I=I+1      Else I=1
    divisor=divisor+salto(I)
    Resto=numero Mod divisor
    If Resto=0 Then
            If numero > divisor Then
            Print "Numero divisible por", divisor
            Sleep
            End
            Else
            Print "Numero primo", numero
```

```
                    Sleep
                    End
                    EndIf
            End If
        Wend
        Print "Numero primo:", numero
        Sleep
        End
```

Veamos a continuación unas cuantas propiedades de la divisibilidad de enteros. Si un número *n* divide a otro *m* utilizaremos la convención $n|m$.

El conjunto **Z** de los números enteros forma un anillo respecto de la suma y el producto.

Propiedad. Sean *n, m, r* números enteros. Si $n|m$ y $m|r \Rightarrow n|r$.

Demostración:

Sea $r = m.q_1$ y $m = n.q_2$. Tenemos pues que $r = n.q_1 q_2$, con lo cual *r* es un múltiplo de *n* y este último lo divide.

Propiedad. Sean *n, m, r* números enteros. Si $n|m$ y $n|r \Rightarrow n|r \pm m$.

Demostración:

Sea $r = n.q_1$ y $m = n.q_2$. Tenemos pues que $r \pm m = n(q_1 \pm q_2)$, con lo cual $r \pm m$ es un múltiplo de *n,* y este último lo divide.

Propiedad. Sean *n, m* números enteros. Si $n|m \Rightarrow n|\lambda.m \forall \lambda \in Z$.

Demostración:

Sea $m = n.q_1$. Tenemos pues que $\lambda.m = \lambda.n.q_1 \Rightarrow n \mid \lambda.m$.

Propiedad. Sea *p* un número primo. Si *p* divide a *n* y $q|n \Rightarrow p.q|n$.

Máximo común divisor (m.c.d.).

Se denomina así al mayor de los divisores comunes a todos los números a que se aplica. Formalmente, se define el m.c.d. de dos números n_1 y n_2, al número $d \geq 0$ tal que $d| n_1$ y $d| n_2$ y se cumple además que *d* es el mayor número de todos los que cumplen esta condición.

Propiedad. Si *p* es primo entonces el mcd(*p*,*n*) = *p* siempre que *p* divida a *n* y 1 en cualquier otro caso.. Como consecuencia de esto se dice que dos números son primos entre sí, si cumplen que su m.c.d. es la unidad.

Propiedad. Sea *p* un número primo y *n* un número entero, si *p* no divide a *n* se cumple que mcd(p^m,*n*) = 1 para todo *n* mayor o igual a 1.

Propiedad. Sean *n, m, r* números enteros con mcd(*n*,*m*) = 1, si se cumple que *n* divide al

producto *m.r*, se cumple que *n* divide a *r*.

Propiedad. Sean *n*, *m*, *r* números enteros, si el mcd(*n*,*m*) = *r*, entonces se cumple que $\mathrm{mcd}\left(\frac{n}{r},\frac{m}{r}\right)=1$.

Propiedad. Sea *p* un número primo tal que p^n divide a *m.r* y tal que *p* no divide al mcd(*m*,*r*), se cumple que o bien p^n divide a *m* o p^n divide a *r*. Nótese que esta proposición es falsa si no se cumple la propiedad de que *p* no divide al mcd(*m*,*r*), sin embargo, si *m* y *r* son relativamente primos no es necesario el cumplimiento de esta propiedad.

Propiedad. Si mcd(*n*,*m*) = 1 y mcd(*n*,*r*) = 1, se cumple que mcd(*n*,*mr*) = 1.

Propiedad. Los divisores comunes a dos números son los comunes al menor de ellos y al resto de la división de ambos. Es decir, sean dos números *a* y *b* con *a* > *b***,** sean *q* el cociente y *r* el resto, tenemos que

$$a = b.q \pm r$$

todo divisor de *a* y *b* lo es de *b.q* y por lo tanto de *r*. Por lo tanto se cumple que el mcd(*a*,*b*)=mcd(*b*,*r*).

De lo anterior se deduce que ya que el mcd(*a*,*b*)=mcd(*b*,*r*), dos números *a* y *b* son primos entre sí cuando lo son *b* y *r*.

Propiedad. Si un número *n* es divisor de un producto de varios factores $n_1, n_2, .., n_r$, es divisor de por lo menos uno de estos factores.

Demostración: Si *n* no es divisor de ninguno de estos factores, será primo con todos ellos, con lo cual su m.c.d. será 1, siendo 1 también el m.c.d del producto de ellos con respecto al número *n*, lo que contradice nuestra afirmación inicial de que *n* es divisor del producto. Con lo cual deducimos que debe ser divisor de alguno de sus miembros.

Teorema fundamental de la aritmética.

Todo número está formado por una descomposición en factores primos que es única. Formalmente, sea *n* >1, *n* admite una descomposición única del tipo $n = p_1^{n_1} p_2^{n_2} .... p_k^{n_k}$ con $n_1, n_2, .., n_k$ enteros mayores que cero y $p_1, p_2, .., p_k$ primos.

Decimos que un número primo *p* es un factor de *n*, si se cumple que *p* es uno de los p_i definidos en el teorema anterior.

Demostración.

Supongamos que existe más de una descomposición canónica del número *n*. Tenemos pues que $n = p_1^{e_1} .... p_k^{e_k} = q_1^{r_1} ... q_s^{r_s}$ con $p_1 < ... < p_k$ y $q_1 < .. < q_s$ primos y $e_1, ..., e_k, r_1, .., r_s$

números enteros positivos. Si tomamos la parte izquierda de la igualdad tenemos que $p_1 | n$, pero $n = q_1^{r_1} ... q_s^{r_s}$, con lo cual p_1 debe dividir a alguno de los q_i, pero esto no es posible ya que los q_i son primos. Con lo cual p_1 es igual a algún q_i. Siguiendo con este razonamiento podemos sustituir todos los p_j por algún q_i con lo que tenemos la igualdad de p_j en ambos lados, pero con distintos exponentes. Sin embargo, para que se cumpla la igualdad los exponentes de ambas partes deben ser iguales, con lo que se cumple que la factorización debe ser única.

Teorema. Sea $n = p_1^{n_1} p_2^{n_2} .... p_k^{n_k}$ la descomposición canónica de un número *n*. Entonces se cumple que todos los divisores de *n* son de la forma $d = p_1^{\alpha_1} p_2^{\alpha_2} ... p_m^{\alpha_m}$ con $0 \le \alpha_i \le n_i$.

Propiedad. Si *p* es primo y $p | ab \Rightarrow p | a$ o $p | b$ o *p* divide a ambos.

Demostración:

Sea $a = p_1^{a_1} p_2^{a_2} .... p_k^{a_k}$ y $b = q_1^{b_1} q_2^{b_2} .... q_m^{b_m}$ tenemos pues que $a.b = p_1^{a_1} p_2^{a_2} .... p_k^{a_k} q_1^{b_1} q_2^{b_2} .... q_m^{b_m}$ con lo que obtenemos que $p | ab \Leftrightarrow p = p_i$ con $i \in [1, k]$ o $p = q_j$ con $j \in [1, m]$.

Definición. Sea *n* un número cuya descomposición canónica es de la forma $n = p_1^{n_1} p_2^{n_2} .... p_k^{n_k}$ se define $\tau(n) = (n_1 + 1)...(n_k + 1)$ como el número de divisores de *n* y $\sigma(n)$ como la suma de los divisores de *n*.

Teorema. Si el mcd(*n*,*m*)=1, entonces $\sigma(a.b) = \sigma(a).\sigma(b)$.

Calculo del máximo común divisor.

El cálculo del máximo común divisor de dos números es una tarea relativamente sencilla. Para calcular el m.c.d. solo tenemos que escoger el menor de los dos números e ir probando con todos los números positivos inferiores a él hasta encontrar uno que divida a ambos números. Este algoritmo evidentemente funciona, sin embargo dista mucho de ser eficiente. Un algoritmo mucho más eficiente es el algoritmo de Euclides, que presentamos a continuación.

Algoritmo de Euclides.

Este algoritmo permite determinar el m.c.d. de dos números *a* y *b*, suponemos que *a>b>0*. En este caso por la propiedad anterior sabemos que $a = b.q_1 \pm r_1$, si el resto r_1 es mayor que cero, hacemos $b = q_2.r_1 \pm r_2$, si el resto sigue siendo mayor que cero, se sigue el proceso hasta que se obtenga un división exacta. Cuando esto se produce, el resto de la operación anterior es el m.c.d. de los números *a* y *b*. Esquemáticamente hacemos:

$$a = b.q_1 \pm r_1$$

$$b = q_2.r_1 \pm r_2$$
$$\ldots\ldots\ldots$$
$$r_{i-2} = q_i.r_{i-1} + r_i$$
$$\ldots\ldots\ldots$$
$$r_{n-2} = q_n.r_{n-1} + r_n$$
$$r_{n-1} = q_{n+1}.r_n$$

En este momento sabemos que r_n es el m.c.d.

Veamos un ejemplo. Supongamos que queremos encontrar el m.c.d. entre *a* y *b*, siendo *a*=15234 y *b*=6532. Hacemos:

$$\begin{aligned} 15234 &= 2*6532 + 2170 \\ 6532 &= 3*2170 + 22 \\ 2170 &= 98 * 22 + 14 \\ 22 &= 1 * 14 + 8 \\ 14 &= 1 * 8 + 6 \\ 8 &= 1 * 6 + 2 \\ 6 &= 3 * \mathbf{2} + 0 \end{aligned}$$

Con lo que tenemos que el mcd(15234, 6532) = 2.

Una versión del mismo obtenida de [BRA90] es la que presentamos a continuación. Este algoritmo emplea un tiempo del orden del logaritmo de sus argumentos.

Algoritmo Euclides(entrada: *a* salida:*b*)
inicio
    Mientras $a > 0$ hacer
        $t \leftarrow b \bmod a$;
        $b \leftarrow a$;
        $a \leftarrow t$;
    Fin Mientras;
    Devolver (*b*);
Fin Euclides;

Algoritmo extendido de Euclides.

El siguiente algoritmo se utiliza para calcular el m.c.d. de dos números *a* y *b*, así como de los valores x e y tales que $ax + by = c$, siendo *c* el mcd(*a*,*b*).

Algoritmo Euclidesext(entrada: *a*,*b* salid*a*: *c*,*x*,*y*)
inicio

Si b = 0 entonces
$$\begin{cases} d \leftarrow a; \\ x \leftarrow 1; \\ y \leftarrow 0; \\ \text{devolver}\,(d, x, y); \end{cases}$$

$x_2 \leftarrow 1$;
$x_1 \leftarrow 0$;
$y_2 \leftarrow 0$;
$y_1 \leftarrow 1$;
Mientras $b > 0$ hacer
Inicio
$q \leftarrow \left\lfloor \frac{a}{b} \right\rfloor$;
$r \leftarrow a - q.b$;
$x \leftarrow x_2 - q.x_1$;
$y \leftarrow y_2 - q.y_1$;
$a \leftarrow b$;
$b \leftarrow r$;
$x_2 \leftarrow x_1$;
$x_1 \leftarrow x$;
$y_2 \leftarrow y_1$;
$y_1 \leftarrow y$;
Fin Mientras;
$d \leftarrow a$;
$x \leftarrow x_2$;
$y \leftarrow y_2$;
devolver(d, x, y);
Fin Euclidesext;

Una versión especializada del algoritmo presentado en [DEN83] puede aplicarse al calculo de inversos modulo *n*. Es decir, resolver la ecuación $ax \bmod n = 1$.

Algoritmo inv(entrada: a salida: n)
Inicio
$g_0 \leftarrow n$;
$g_1 \leftarrow a$;
$u_0 \leftarrow 1$;
$v_0 \leftarrow 0$;
$u_1 \leftarrow 0$;
$v_1 \leftarrow 1$;
$i \leftarrow 1$;
Mientras $g_i \neq 0$ hacer
Inicio

$y \leftarrow \left\lfloor \frac{g_{i-1}}{g_i} \right\rfloor;$

$g_{i+1} \leftarrow g_{i-1} - y.g_i;$

$u_{i+1} \leftarrow u_{i-1} - y.u_i;$

$v_{i+1} \leftarrow v_{i-1} - y.v_i;$

$i \leftarrow i+1;$

Fin mientras;

$x \leftarrow v_i - 1;$

Si $x \geq 0$ entonces devolver(x) sino devolver($x+n$);

Criterio general de divisibilidad. La condición necesaria y suficiente para que un número *m* sea múltiplo de otro *n*, es que contenga todos los factores primos de éste, con iguales o mayores exponentes.

Propiedad. Todo número compuesto *n* es divisible por otro, primo absoluto, cuyo cuadrado no le excede.

Congruencias.

Se dice que *a* es congruente con *b* modulo *n*, si se cumple que la división de ambos por *n* da el mismo resto como resultado. Es decir si *a* y *b* son congruentes modulo *n*, quiere decir que para algún entero *k* se cumple que $a-b=kn$ y se representa por $a \equiv_n b$, $a \equiv b \pmod{n}$ o simplemente $a = b \pmod{n}$.

La relación $a \equiv b \pmod{n}$ es una relación de equivalencia en **Z**. Además se cumple que el conjunto de los enteros modulo *n* forman un anillo conmutativo con respecto a la suma y multiplicación. Formalmente, sea $a \in \mathbf{Z}$ podemos definir la clase de equivalencia de *a* como $\overline{a} = \{a + kn : k \in \mathbf{Z}\}$.

Propiedad. Todo número es congruente consigo mismo, respecto de cualquier módulo.

Propiedad. Dos números congruentes con un tercero, respecto de un mismo módulo, son congruentes entre sí respecto al mismo módulo.

Propiedad. Todos los números múltiplos de *m* son congruentes con cero respecto de dicho módulo.

Propiedad. Sea *a* un número primo con *m*, todo número *b* congruente con *a* modulo *m* es primo con *m*.

Propiedad. La condición necesaria y suficiente para que dos números sean congruentes entre sí respecto de un mismo módulo, es que su diferencia sea un múltiplo de ese módulo.

Propiedad. Se puede multiplicar o dividir los miembros y el módulo por un divisor común, que el resultado se mantiene inalterado.

Propiedad. Si dos números son congruentes respecto de varios módulos, son congruentes respecto del m.c.m. de éstos.

Propiedad. Si el mcd(*a*,*n*) = 1, se cumple que $a.i(\mathrm{mod}\, n) \neq a.j(\mathrm{mod}\, n) \forall i, j$ tales que $0 \leq$ i < j < n.

Demostración. Supongamos que *n* divide a *a.(i - j)*.Tenemos pues que *(i - j)* debe ser un múltiplo de *n*, ya que el mcd(*a*,*n*) = 1, pero eso es imposible ya que *i*, *j* son menores que *n*.

Una consecuencia importante de la propiedad anterior es que *a.i* (mod *n*) es una permutación del conjunto completo de residuos $\{0,...,n-1\}$.

Teorema de invertibilidad. Sean *a*, *n* números enteros tales que se cumple que el mcd(*a*,*n*)=1. Existe un único entero *x*, con 0 < *x* < *n*, tal que $a.x(\mathrm{mod}\, n) = 1$.

Demostración. Por la propiedad anterior tenemos que el conjunto formado por todos los elementos de la operación *a.i* (mod *n*), con *i*=0,1,2,…,*n* – 1 es una permutación del conjunto $\{0,...,n-1\}$, y por lo tanto existirá un elemento que hará que *a.i* (mod *n*) = 1.

Operaciones con congruencias.

Propiedad. La suma o resta de varias congruencias modulo *m* da como resultado otra congruencia respecto del mismo modulo.

Propiedad. El producto de varias congruencias respecto de un módulo común *m* da como resultado otra congruencia respecto del mismo módulo.

Como consecuencia de esto tenemos que si $a \equiv n.b \bmod m$ entonces $a.n \equiv n.b.n \bmod m$.

Propiedad. Si en un polinomio $a_0x^n+a_1x^{n-1}+....+a_{n-1}x+a_n$ se sustituyen dos valores, *a* y *b*, de *x*, congruentes modulo *m*, los valores obtenidos son congruentes respecto al mismo módulo.

Como consecuencia tenemos:

- Si un número *a* satisface la congruencia $a_0x^n+a_1x^{n-1}+...+a_{n-1}x+a_n \equiv_n 0$ todo número congruente con *a* módulo *m* cumple la misma condición.
- Los cocientes de dividir dos números congruentes modulo *m* por un divisor común, primo con *m*, son congruentes respecto del mismo módulo.
- Se puede dividir los dos miembros de una congruencia por cualquier divisor común, dividiendo también el módulo por su m.c.d. con el divisor.
- Sean dos números *h* y *k* primos respecto a *m*, se cumple siempre que dados cualesquiera *a* y *b*, $a.h \equiv b.k \bmod m$ es equivalente a $h \equiv k \bmod m$.

Calculo de los restos potenciales.

Sean $r_1, r_2, .., r_n$ los restos modulo *m* de las potencias $a_1, a_2, .., a_n$ para hallar el resto r_{n+1} basta con hallar el resto de $r.r_n \bmod m$.

Indicador de un número.

Dado un número *n*, definimos el indicador de *n* y lo denotamos por $\phi(n)$ como el número de números primos respecto a *n* y menores que él. Al conjunto formado por estos números se le denomina *conjunto reducido de restos*. Consideramos $\phi(1) = 1$.

Propiedad. Sea *p* un número primo, su indicador es *p* - 1, es decir $\phi(p) = p - 1$.

Propiedad. Sean *m* y *n* números enteros con *m* relativamente primo respecto a *n*. Se cumple que *m* es congruente modulo *n* con alguno de los elementos del conjunto reducido de restos modulo *n*.

Demostración. Al ser *m* y *n* relativamente primos, se cumple que $\text{mcd}(m.n) = 1$, con lo que tenemos que para cada numero q_i con $1 \le i \le \phi(n)$, se cumple que el $\text{mcd}(mq_i, n) = 1$ y que $m.q_i \bmod n = q_j$ para algún q_j perteneciente al conjunto reducido de restos. Con lo que tenemos que el conjunto $m.q_i \bmod n = q_j$ es una permutación del conjunto reducido de restos.

Propiedad. El indicador del producto de dos números primos entre sí, es el producto de sus indicadores. Es decir, sea *n* el producto de dos número primos *p* y *q cualesquiera*. Se cumple siempre que $\phi(n) = \phi(p).\phi(q) = (p - 1).(q - 1)$. Esta propiedad se puede generalizar de la siguiente manera, dado un número *n* cualquiera, la función $\phi(n)$ viene dada por la siguiente expresión:

$$\phi(n) = \prod_{i=1}^{n} p_i^{e_i - 1}(p_i - 1)$$

Se considera el número *n* como $n = p_1^{e_1}.p_2^{e_2}..p_t^{e_t}$.

Propiedad. Sea *p* un número primo, se cumple que $\phi(p^\alpha) = p^{\alpha-1}.(p - 1)$.

Lema. Sea *p* un número primo tal que $p > 0$ y sean *a* y *b* enteros se cumple siempre que

$$(a+b)^p \equiv a^p + b^p \bmod p.$$

Demostración.

Por la formula binomial tenemos que $(a+b)^p = a^p + b^p + \sum_{i=1}^{p-1}\binom{p}{i}a^{p-i}b^i$, con lo que probando que $\sum_{i=1}^{p-1}\binom{p}{i}a^{p-i}b^i \equiv 0 \bmod p$ tenemos demostrado el lema. Por la definición de binomial tenemos que $\binom{p}{i} = \frac{p(p-1)..(p-i+1)}{i!}$, al ser $i < p$ tenemos que $\binom{p}{i}$ es un múltiplo de *p* con los que se cumple que $\sum_{i=1}^{p-1}\binom{p}{i}a^{p-i}b^i \equiv 0 \bmod p$.

La verdad es que no hay que asustarse mucho al ver las formulas anteriores. Podemos resumir el cálculo de $\phi(n)$ de la siguiente manera [RAM99]:

1. n es primo entonces $\phi(n) = n - 1$. Por ejemplo si $n = 7$ entonces $\phi(7) = 7 - 1 = 6$.
2. n es del tipo

Pequeño teorema de Fermat (Congruencia de Fermat).

Sea *p un número primo y a* un número entero, se cumple siempre que $a^p \equiv a \bmod p$.

Demostración.

La demostración la realizaremos por inducción. Evidentemente se cumple que $1^p \equiv 1 \bmod p$, suponemos pues que $n^p \equiv n \bmod p$ y debemos demostrar que $(n+1)^p \equiv n+1 \bmod p$. Por el lema anterior tenemos que $(n+1)^p \equiv (n^p + 1^p) \equiv (n^p + 1) \bmod p$ por la hipótesis de inducción tenemos que $n^p \equiv n \bmod p$. Sustituyendo tenemos que $(n+1)^p \equiv n^p + 1 \equiv (n+1) \bmod p$ como queríamos probar.

Si bien el teorema de Fermat es el anterior, se suele conocer más frecuentemente con el enunciado siguiente, que es simplemente una versión del que acabamos de enunciar.

Teorema de Fermat.

Sean a y p dos números primos entre sí, siendo p un número primo, se cumple que $a^{p-1} \equiv 1 \bmod p$.

Una aplicación muy importante del teorema de Fermat es la posibilidad de reducción del problema de calcular el residuo de un número a^k con un k muy grande y tal que $k > p - 1$. Para hacerlo, simplemente partimos del supuesto de que p no es un factor de a, ya que si lo fuera el residuo sería 0. Dividimos pues k por $p - 1$ con lo que podemos representar k como $k = (p-1)q + r \quad 0 \le r < p-1$ y $q, r \ge 0$.

Tenemos pues que $a^k \equiv a^{(p-1)q+r} \equiv \left(a^{p-1}\right)^q a^r \bmod p$, pero por el teorema de Fermat $a^{p-1} \equiv 1 \bmod p$ con lo que tenemos que $a^k \equiv a^r \bmod p$. Es decir podemos reducir el cálculo de los residuos de una manera drástica.

Generalización de Euler.

Sean n y m dos números primos entre sí, se verifica que $n^{\phi(m)} \equiv 1 \bmod m$, en particular si m es primo tenemos la congruencia de Fermat.

Demostración.

Sea $\{r_1, r_2, .., r_{\phi(n)}\}$ el conjunto reducido de restos módulo n, con $0 < r_i < n$ con $1 \le i \le \phi(n)$. En este caso $\{n.r_1 \bmod m, ..., nr_{\phi(m)} \bmod m\}$ es una permutación del conjunto $\{r_1, r_2, .., r_{\phi(n)}\}$. Con lo cual tenemos que $\prod_{i=1}^{\phi(m)} (n.r_i \bmod m) \equiv \prod_{i=1}^{\phi(m)} r_i \Rightarrow (n^{\phi(m)} \bmod n) \prod_{i=1}^{\phi(m)} r_i \equiv \prod_{i=1}^{\phi(m)} r_i \Rightarrow n^{\phi(m)} \bmod n \equiv 1$.

La generalización de Euler nos da un método para resolver ecuaciones del tipo $ax \bmod n = 1$, en las cuales los números *a* y *n* son relativamente primos. La solución en este caso viene dada por $x = a^{\phi(n)-1} \bmod n$, que en el caso de que *n* sea primo se reduce a $x = a^{n-2} \bmod n$.

Algoritmo ecuaciones(entrada: *a*,*n* salida: *b*)
Inicio
$g \leftarrow$ Euclides(a, n);
Si $b \bmod g = 0$ entonces
Inicio
Imprimir ("Solución :", *g*);
$n_0 \leftarrow \frac{n}{g}$;
$x_0 \leftarrow inv(\frac{a}{g}, n_0)$;
$x_1 \leftarrow \left(\left(\frac{b}{g}\right).x_0\right) \bmod n$;

Desde $t \leftarrow 0$ hasta $t - n$ hacer
Inicio
$x \leftarrow (x_1 + t.n_0) \bmod n$;
imprime(x);
Fin Desde;
Sino imprime ("No hay soluciones);
Fin Si;
Fin ecuaciones;

Teorema chino del resto.

Sean $p_1, p_2, .., p_k$ números primos entre sí, es decir, $\text{mcd}(p_i, p_j) = 1$ para $i \neq j$, y sea $n = p_1.p_2..p_k$. Se cumple que el sistema $x \equiv x_i \bmod p_i$ con $i = 1,..,k$ tiene solución única $x \in [0, n-1]$.

Demostración:

Se definen $P_1,..,P_k$ como $n = p_1.p_2...p_k = p_1.P_1 = p_2.P_2 = ... = p_k.P_k$. Ya que $\text{mcd}(P_i, p_i) = \text{mcd}\left(\frac{n}{p_i}, p_i\right) = 1 \ \forall i \in 1..k$ sabemos que existe una solución única de la ecuación $P_j.x \equiv 1 \bmod p_j$. Supongamos que la solución a esa ecuación es P_j^{-1}, si creamos el número $x' = \sum_{i=1}^{k} P_i P_i^{-1} x_i$, tenemos que $x' \equiv P_i P_i^{-1} x_i \equiv x_i \bmod p_i$ es una solución para cada *i*, ya

que por definición $P_i P_i^{-1} \equiv 1 \bmod p_i$ y $P_i P_i^{-1} \equiv 0 \bmod p_j$. Con lo que tenemos que x' es la solución general del sistema.

El siguiente programa calcula la solución a un sistema de ecuaciones mediante teorema chino del resto. Para hacerlo utiliza la rutina para calcular inversos modulo n definida anteriormente.

Algoritmo chino(entrada: n, $p_1,..,p_t, x_1,..x_t$ salida: x)
Inicio

Desde $i \leftarrow 1$ hasta t hacer $y_i \leftarrow inv\left(\left(\frac{n}{d_i}\right) \bmod d_i, d_i\right)$;

$x \leftarrow 0$;

Desde $i \leftarrow 1$ hasta t hacer $x \leftarrow \left(x + \left(\frac{n}{d_i}\right) y_i x_i\right) \bmod n$;

Devolver(x);
Fin chino;

Teorema de Wilson.

Este teorema atribuido a Wilson, aunque parece ser que ya había sido publicado por Waring, afirma que $(p-1)! \equiv -1 (\bmod p)$. Un par de consecuencias interesantes del teorema de Wilson son las siguientes:

a) Un entero $n > 1$ es primo $\Leftrightarrow (n-1)! \equiv -1 (\bmod n)$. O lo que es lo mismo n es primo si divide exactamente a $((n-1)!+1)$.
b) Si p es primo tal que $p \equiv 1 (\bmod 4)$, se cumple que la congruencia $x^2 \equiv -1 (\bmod p)$ tiene como soluciones $x = \pm((\frac{1}{2}(p-1)))!$.

Algoritmo de exponenciación rápida.

Propuesto por Denning en [DEN88]. Se trata de un algoritmo basado en la aplicación de sucesivos pasos de obtención de raíces cuadradas y multiplicaciones. El tiempo de ejecución T está acotado por $\lfloor \log_2 z \rfloor + 1 \le T \le 2\lfloor \log_2 z \rfloor + 1$. El programa devuelve el valor de $a^z \bmod n$.

Función rapidexp (entrada: a, z, n salida: x)
Inicio

$a_1 \leftarrow a$;

$z_1 \leftarrow z$;

$x \leftarrow 1$;

Mientras $z_1 \neq 0$ hacer

Inicio

Mientras $z_1 \bmod 2 = 0$ hacer
Inicio
$z_1 \leftarrow \left[\frac{z_1}{2}\right]$;
$a_1 \leftarrow (a_1.a_1) \bmod n$;
Fin mientras;
$z_1 \leftarrow z_1 - 1$;
$x \leftarrow (x.a_1) \bmod n$;
Fin mientras;
Devolver (*x*);
Fin rapidexp;

Propiedades de los números primos.

La primera propiedad importante referente a los números primos es la existencia de infinitos de ellos. Es decir el conjunto de los números primos es infinito. La demostración de esta afirmación es de Euclides y es más o menos como sigue. Supongamos que el conjunto de números primos es finito y está formado por los números $p_1,p_2,...p_n$, formamos un número $P=p_1.p_2...p_n + 1$, este número o bien es primo con lo cual contradice nuestra afirmación inicial de que el conjunto de primos es finito o bien admite un divisor primo *q* diferente de $p_1,..., p_n$ ya que el resto de dividir *P* por cualquiera de ellos es 1.

Existen algunos resultados interesantes relativos a los números primos. Por ejemplo se sabe que $p_n < 2^{2^n}$, es más se sabe también que $p_n \sim n \log n$ cuando $n \to \infty$. El Teorema del número primo afirma que para cualquier valor x el número de primos $\pi(x) \sim \frac{x}{\log x}$ cuando $x \to \infty$. Los números del tipo $2^n - 1$ que son primos se denominan primos de Mersenne en honor a Marin Mersenne un clérigo del siglo XVII aficionado a las matemáticas. Estos son muy útiles para proporcionar números primos grandes, es más el mayor número primo que se conoce hasta el momento es el número de Mersenne M(3021377).

Un conjunto de números con propiedades similares son los números de Fermat. Estos son los números de la forma $2^{2^n} + 1$. Fermat consideraba que todos los números de esta forma eran primos, sin embargo más tarde se demostró que F(5) era compuesto.

Otro conjunto de números interesantes es el de los denominados perfectos. Estos son todos aquellos números tales que la suma de sus factores anteriores a él dan como resultado el mismo número. Evidentemente ningún número primo puede ser perfecto, sin embargo, Euclides sabía que los números de la forma $2^{n-1}(2^n - 1))$ eran perfectos si $2^n - 1$ era primo. Fue sin embargo Euler en el siglo XVIII el que probó que todos los números perfectos pares tenían esa forma.

Otra forma sencilla de encontrar números primos es la de utilizar la denominada formula primorial. Se define como primorial de un número *p* y se identifica como $p^{\#}$ al producto de todos los números primos menores o iguales que *p*. Un número primo de la forma $p^{\#} + 1$ se denomina un número primo primorial. Por el momento solo se han conseguido encontrar 16 primos de este tipo, siendo el mayor de ellos correspondiente a $p = 24027$ con

10387 dígitos[COU99].

Otra interesante propiedad es la de que existen infinitos números primos de la forma 4*r*-1. La demostración es simple, parte del hecho de que todos los números primos, exceptuando el 2 tienen la forma 4*r*-1 o 4*r*+1. Sean $p_1, p_2, .., p_k$ primos de la forma 4*r*-1, ninguno de ellos igual a 3. Sea $n = 4p_1p_2..p_k + 3$. Entonces *n* factoriza en primos, ninguno de los cuales puede ser 2 ya que el número es impar. Supongamos que todos los p_i son de la forma 4*r*+1, entonces n tendría también esa forma, sin embargo por la definición anterior vemos que eso no es cierto, debe pues haber algún primo de la forma 4*r*-1. Este no puede ser ninguno de los p_i, con lo cual queda demostrada la infinidad de los números de la forma 4*r*-1.

La conjetura de Goldbach dice que cada entero par mayor que dos es la suma de dos números primos. Otra famosa conjetura es la de que existen infinitos pares de primos cuya diferencia es dos. Ninguna de ellas ha podido ser probada hasta el momento. Un teorema importante es el de Vinogradov que dice que *cada entero impar suficientemente grande es la suma de tres primos.*

En cuanto a su distribución, citemos el postulado de Bertrand que dice que para cada número *n* > 1, existe siempre un número primo entre *n* y 2*n.*

Existen algunos polinomios que nos permiten generar números primos. Entre ellas podemos citar las siguientes dadas por Euler:

$$x^2 + x + 17 \text{ da números primos en el intervalo } [0,16]$$
$$2x^2 + 29 \quad \text{da números primos en el intervalo } [0,28]$$
$$x^2 + x + 41 \text{ da números primos en el intervalo } [0,40]$$

Estas funciones podrían hacer pensar en la existencia de polinomios capaces de generar los números primos. Sin embargo, esto no es cierto, tal como muestra el siguiente teorema:

Teorema. Dado un polinomio *f(x)* con coeficientes enteros, existen infinitos números positivos *c* tales que *f(c)* es un número compuesto. Una demostración limitada para polinomios cuadráticos puede encontrarse en [COU99].

Determinación del número de primos en un intervalo dado.

Se define la función $\pi(x)$ como el número de primos menores o iguales a *x*. Hay que hacer notar que se considera como primer número primo el 2, con lo cual $\pi(2) = 1$. La forma más sencilla de calcular $\pi(x)$ es simplemente contando el número de primos hasta el número dado. Evidentemente esta manera de calcular $\pi(x)$ no es práctica. Afortunadamente existen otras maneras de calcularlo mediante formulas que dan una aproximación lo suficientemente exacta de $\pi(x)$, una de ellas fue enunciada por Legendre y es como sigue:

$$\pi(x) + 1 = \pi(\sqrt{x}) + [x] - \sum_{p_i \le \sqrt{x}} \left[\frac{x}{p_i} \right] + \sum_{p_i \le p_j \le \sqrt{x}} \left[\frac{x}{p_i p_j} \right] - \sum_{p_i < p_j < p_k \le \sqrt{x}} \left[\frac{x}{p_i p_j p_k} \right] + \ldots$$

Desgraciadamente esta formula no mejora mucho el esfuerzo de cálculo necesario para

calcular $\pi(x)$ con respecto a la simple enumeración, lo que la hace impracticable para tamaños grandes de *x*.

Una mejora sustancial fue dada por Meissel, cuya formula es una modificación de la de Legendre para hacerla más eficiente.

En la tabla siguiente se especifican el número de primos en intervalos de 1.000.000 hasta 200.000.000. Para calcularlos se ha utilizado el programa I especificado en el apéndice de programas. La tabla se lee de la siguiente manera, la celda especificada por la intersección de la fila y columna marcada como 10 y 5 respectivamente indicará el número de primos en el intervalo [14.000.000, 14.999.999].

Millones	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	78498	70434	67882	66329	65366	64335	63798	63128	62711	62089
10	61937	61542	61191	60824	60626	60425	60183	60052	59682	59556
20	59335	59317	58959	58900	58804	58599	58537	58364	58245	58182
30	58119	57835	57851	57711	57395	57486	57360	57342	57435	57251
40	57101	56863	56914	56848	56775	56892	56639	56450	56386	56602
50	56359	56348	56208	56150	55996	56129	56104	55900	55977	55800
60	55929	55554	55705	55779	55467	55568	55643	55574	55331	55389
70	55308	55284	55430	55164	55049	55306	54923	55008	54899	54937
80	55026	55020	54886	54821	54591	54738	54709	54651	54732	54388
90	54704	54577	54443	54422	54644	54451	54363	54430	54126	54331
100	54207	54315	54303	54205	54206	54070	54060	54134	54214	54130
110	53931	53815	54003	53791	53796	53912	53882	53771	53677	53751
120	53618	53768	53635	53807	53661	53671	53612	53607	53697	53453
130	53517	53483	53362	53381	53526	53267	53478	53281	53337	53370
140	53100	53316	53431	53354	53188	53192	53300	53117	53149	53040
150	52995	53045	52978	53159	52975	53071	53059	52958	52804	53008
160	53053	52716	52913	53113	52724	52681	52883	52915	52748	52869
170	52737	52855	52702	52791	52599	52786	52710	52764	52554	52794
180	52393	52501	52314	52648	52740	52573	52436	52559	52391	52523
190	52298	52254	52533	52464	52550	52320	52225	52153	52295	52361

Teorema de los números primos.

Este teorema, conjeturado por Gauss aunque generalmente atribuido a Riemann, dice que cuando *x* es muy grande, $\pi(x)$ se aproxima a $\frac{x}{\log x}$. Más formalmente

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1.$$

Números de Carmichael.

Un número *n* se dice que es un número de Carmichael si se cumple que:

1) $n > 0$ es un numero compuesto, impar

2) $b^n \equiv b \bmod n$ para todos los enteros *b*.

Existen infinitos números de Carmichael, siendo el más pequeño de ellos el 561. Una caracterización más completa de los números de Carmichael viene dada por el teorema de Korselt.

Teorema de Korselt.

Un entero $n > 0$ impar es un número de Carmichael si y solo si se cumplen las siguientes condiciones para cada *p*, factor primo de *n*:

1) p^2 no divide a *n*
2) $p - 1$ divide a $n - 1$.

Restos cuadráticos.

Sea *p* un número primo y *x* un entero tal que $1 \le x \le p-1$, se dice que *x* es un resto cuadrático módulo *p* si la congruencia $y^2 \equiv x \bmod p$ tiene una solución $y \in Z_p$. En caso contrario se llama no-resto cuadrático.

Criterio de Euler.

Sea *p* un número primo, *x* es un resto cuadrático módulo *p* si y solo si $x^{(p-1)/2} \equiv 1 \bmod p$.

Demostración.

Supongamos que *x* es un número primo distinto de cero, entonces por el teorema de Fermat sabemos que $x^{p-1} \equiv 1 \bmod p$, con lo que tenemos que

$$x^{(p-1)/2} \equiv (y^2)^{(p-1)/2} \bmod p \equiv y^{p-1} \bmod p \equiv 1 \bmod p.$$

Inversamente, si $x^{(p-1)/2} \equiv 1 \bmod p$, suponemos que *b* sea un elemento primitivo módulo *p*. Tenemos que $x^{b^i} \bmod p$ para algún *i*. Tenemos entonces que

$$x^{(p-1)/2} \equiv (b^i)^{(p-1)/2} \bmod p \equiv b^{i(p-1)/2} \bmod p.$$

Como tenemos que *b* debe tener orden *p* - 1, *i* debe ser par, con lo que las raíces cuadradas de *x* son del tipo $\pm b^{i/2}$.

Test de primalidad.

El método más simple para determinar si un número es primo o no es la criba de Erastotenes. Este método consistía en poner en una tabla todos los números en el intervalo en el que se pretendía encontrar los primos y luego se iban tachando por orden todos los números

que fueran múltiplos de cada uno de los números no tachados de la tabla. Los números que permanecían sin tachar eran los primos de ese intervalo. La función de este algoritmo no es en sí la determinación de si un número es primo o no, sino, más bien la de determinar todos los números primos de un intervalo. Sin embargo la idea de dividir un número por todos los primos inferiores a él para determinar la primalidad del mismo es un método sencillo, aunque no eficiente.

Generalmente se suele utilizar unas pequeñas propiedades suplementarias para dar más velocidad a la criba de Erastotenes. La primera consiste en dividir el número por todos los primos anteriores a él en el intervalo $\left[2,\sqrt{n}\right]$, ya que si existe la factorización estará formada por un número mayor y uno menor a $\sqrt{n}$ a no ser que el número sea un cuadrado, en cuyo caso los dos factores son iguales a $\sqrt{n}$. Otra ventaja la podemos obtener si nos fijamos en que, como consecuencia de la propiedad anterior, si *p* es primo, el primer número compuesto que no sea divisible por los primos anteriores a *p* debe ser mayor que p^2. Con estas dos mejoras puede implementarse un algoritmo que nos permita decidir si un número *n* es primo o no. En el caso de no encontrar ninguno de ellos que sea divisor de *n* podemos deducir que *n* es primo. Evidentemente este método no sirve para números grandes como los utilizados en los sistemas de clave pública.

Test de Fermat.

Por el teorema de Fermat sabemos que si *a* y *p* son números primos entre sí, siendo *p* un número primo, se cumple que $a^{p-1} \equiv 1 \bmod p.$ Supongamos que *n* es un número impar compuesto que satisface la ecuación $a^{n-1} \equiv 1 \bmod n$ para algún entero *a* tal que $1 < a < n-1$, este número no tiene por que ser primo, aunque la mayoría de las veces se cumple que si lo es. Estos números son denominados pseudoprimos de la base *a.*

Una posible forma de determinar la primalidad de un número sería pues aplicar el teorema de Fermat a todas las bases entre 1 y *n* –1, sin embargo, el esfuerzo computacional es incluso mayor que en el caso anterior.

La mayoría de los métodos actuales son probabilísticos y determinan la primalidad de un número en función de una probabilidad tan pequeña como se quiera de que éste sea compuesto.

Test de Solovay-Strassen.

Se trata de un método probabilístico en el que lo que se determina es que un determinado número *n* es primo con una probabilidad muy alta. Se escoge un número *n* aleatoriamente de forma que no sea par ni divisible por 3, por 5 ni por 11. Esto último se sabe comprobando que la suma de los dígitos pares y los impares no sea igual modulo 11. La comprobación de que el número es primo es como sigue:

1) Se escogen cien números $a_1,..,a_{100}$ aleatoriamente en el intervalo [1, *p* - 1].

2) Si para cada número a_i, el $\text{mcd}(p, a_i) = 1$ y $\left(\frac{a}{p}\right) \bmod p = a^{(p-1)\cdot/\ 2} \bmod p$ donde $\left(\frac{a}{b}\right)$ es el símbolo de Jacobi para el par *a*, *b* definido como el símbolo de Legendre si *b* es primo:
$\left(\frac{a}{b}\right) = \begin{cases} +1 \text{ si } a \text{ es un residuo cuadratico de } b \\ -1 \text{ en caso contrario} \end{cases}$

Y, si *b* no es primo, pero tiene una factorización en primos $b = p_1.p_2...p_k$, entonces
$\left(\frac{a}{b}\right) = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\left(\frac{a}{p_3}\right)...\left(\frac{a}{p_k}\right)$

Un número *p* que pasa el test tendrá una probabilidad de *no ser primo* de aproximadamente 1 entre 2^{200}. Esto es así ya que si *p* es primo, los valores del símbolo de Jacobi para *p* y a_i coincidirán siempre con $p^{(a_i-1)/2}$, por definición de residuo cuadrático y del símbolo de Jacobi. Por otra parte, si *p* no es primo, la probabilidad de que el símbolo de Jacobi y el exponente coincidan es de $\frac{1}{2}$. Consecuentemente, la probabilidad de que un número *p* no sea primo y pase el test es de $\frac{1}{2}$; la probabilidad de que un número que no es primo puede pasar el test es de $\left(\frac{1}{2}\right)^{100}$.

Test de Lehmann y Peralta.

Se trata de un test mucho más sencillo que el anterior. Fue desarrollado por Lehmann y posteriormente de forma independiente por Peralta. En este caso los pasos a seguir son:

1) Se escogen 100 números aleatorios a_i, con *i* = 1,..,100 en el intervalo [1, *p* - 1].
2) Si $a_i^{(p-1)/2} \equiv 1$ para todos los *i* =1,..,100 entonces *p* es compuesto.
3) Si $a_i^{(p-1)/2} \neq 1$ o –1 para algún *i* =1,..,100 entonces *p* es compuesto.
4) Si $a_i^{(p-1)/2} \equiv 1$ o –1 para todos los *i* =1,..,100, y $a_i^{(p-1)/2} \equiv -1$ para algún *i* =1,..,100 entonces *p* es primo.

Test de primalidad de Miller.

Si bien no se trata de un algoritmo probabilístico, si que es la base del algoritmo probabilístico más ampliamente utilizado, el de Miller-Rabin. El test de Miller es un algoritmo muy sencillo introducido por G. L. Miller en 1976. Básicamente el algoritmo trata de encontrar la mayor potencia de dos que divide a *n* – 1 y su factor *m*. Si *n* es un número primo, entonces se cumple:

a) $a^{n-1} \equiv 1 \bmod n$ por el teorema de Fermat.
b) Si *a* es primo, entonces sabemos que $a^{n-1} \equiv a^{2^k m} \equiv 1 \bmod n$ ya que *n* – 1 es par,

con lo que ya que si $x^2 \equiv 1 \bmod n \Rightarrow x \equiv \pm 1 \bmod n$ cuando *n* es primo, entonces no puede haber ningún residuo que sea diferente de 1 o –1 $(-1 \bmod n \equiv n-1 \bmod n)$.

Basándonos en las anteriores propiedades podemos deducir que o bien el primer elemento es 1 o alguno de los residuos restantes será igual a *n* –1. En el caso de que no se cumpla ninguno de los dos casos, *n* es sin duda compuesto, es pues un algoritmo para determinar si un número es compuesto. Si el algoritmo no determina que el número es compuesto no podemos garantizar con absoluta fiabilidad que sea primo. El algoritmo es pues como sigue:

1) Seleccionar un número impar $n > 1$ y un entero aleatorio *b* relativamente primo a *n*, que denominaremos la base, tal que $1 \le b \le n-1$.
2) Calcular $k = n-1$ y $r = b^k \bmod n$.
3) Si $r \equiv 1 (\bmod n)$ continuamos, en caso contrario el test falla y acabamos el proceso.
4) Ejecutar mientras k sea par y, $r \equiv 1 (\bmod n)$.
   Calcular $k = k/2$ y, $r = b^k \bmod n$.
5) Si *r* = 1 o *r* = *n* -1 entonces *n* falla el test, en caso contrario el número es compuesto.

Si el test falla el número es compuesto, sin embargo, que el número supere el test no es una garantía de primalidad, puede ocurrir que un número compuesto supere el test. Estos números se denominan *pseudoprimos fuertes* para la base *b*.

Test de primalidad de Miller-Rabin.

Se trata de un algoritmo probabilistico basado en el test anterior y en el teorema de Rabin, que dice:

Teorema de Rabin.

Sea $n > 0$ un entero impar. Si el test de Miller aplicado a *n* falla el test en más de $\frac{n}{4}$ bases entre 1 y $n-1$, el número *n* es primo.

Basándonos en el teorema anterior podemos deducir que la probabilidad de que un número impar que no es primo pueda pasar el test es de $\frac{1}{4}$, con lo que si aplicamos el test a *k* bases diferentes la probabilidad de que un número no primo pueda pasar el test es de $\left(\frac{1}{4}\right)^k$. Este resultado no permite determinar con una probabilidad tan alta como queramos que un número no es compuesto. En general se suele utilizar como bases los primeros números primos, con 40 bases la probabilidad de error es inferior a 10^{-20} [COU99]. El algoritmo sería el siguiente:

1) Seleccionar un número impar $n > 0$ y un entero aleatorio *b*, que denominaremos la base, tal que $1 \le b \le n-1$.

2) Calcular por división reiterada de $n - 1$ por 2 los números k y m tales que se cumple $n-1 = 2^k m$, donde m es impar.
3) Calcular $r = b^m \bmod n$.
4) Si $r \equiv 1 (\bmod n)$ el número es primo y acabamos el proceso.
5) Ejecutar k-1 veces
6) Si $r \equiv 1 (\bmod n)$ el número es primo y acabamos el proceso, en caso contrario calculamos $r = r^2 \bmod n$.
7) El número es compuesto.

Debemos hacer notar que aunque escojamos una probabilidad de error muy baja, existen números, particularmente los de Carmichael, que pasan el test sin ser primos. Un resultado de los trabajos de Alford, Granville y Pomerance es que:

Dado cualquier número de bases finito, existen infinitos número de Carmichael que son pseudoprimos fuertes para todas esas bases.

Métodos de factorización.

En el apartado anterior determinamos si un número era primo o no con una probabilidad tan elevada como sea necesario. Pasamos ahora a estudiar los métodos para determinar los números cuyo producto es el número buscado. La determinación de si un número era compuesto tal como la habíamos definido era simple. Un número *n* era compuesto si existían dos números *p* y *q* tales que $n = p.q$. Evidentemente por definición de números primos y compuestos, un número es compuesto si no es primo, con lo que, para determinar si un número es compuesto, basta con determinar que no es primo.

Otra definición de número compuesto basada en el teorema de Fermat es como sigue[COU99]:

Sea $n > 0$ un número entero primo. Si existe un entero b tal que

1) $1 < b < n-1$, y
2) $b^{n-1} \not\equiv 1 \bmod n$,

se cumple que *n* es compuesto.

Los problemas de este método son dos, primero, encontrar el número *b*, que se suele denominar testigo, que nos permitirá determinar si el número es compuesto, y finalmente que la determinación del testigo no nos dará ninguna indicación de los factores que lo forman.

Factorización por división.

Se trata de un método muy sencillo y que puede ser adecuado en la factorización de números de hasta 12 dígitos. Consiste en la división del número *n* por todos los números impares hasta $\sqrt{n}$. Ni que decir tiene que no es útil en la factorización de grandes números.

Algoritmo de factorización de Fermat.

Al igual que en el caso anterior se trata de un método muy sencillo pero potente desarrollado por Fermat. Este método funciona bien cuando n tiene un factor, no necesariamente primo, no mucho mayor que $\sqrt{n}$. Se basa en la idea de intentar conseguir números enteros positivos x e y tales que $n = x^2 - y^2 = (x+y)(x-y)$. El algoritmos es como sigue:

Algoritmo Fermat(entrada: n salida: $factor1$, $factor2$)
Inicio
$factor1 \leftarrow 0$;
$factor2 \leftarrow 0$;
$x \leftarrow \sqrt{n}$;
Si x es un entero parar, x es un factor de n;
Bucle:
$x \leftarrow x+1$;
Si $x = \frac{n+1}{2}$ parar, n es primo;
$y = \sqrt{x^2 - n}$;
Si y es un entero parar, $x + y$ y $x - y$ son factores de n
Sino ir al Bucle;
Fin Fermat;

El método p-1 de Pollard.

Se trata de un algoritmo sencillo, pero de gran potencia. Fue desarrollado en 1974 por Pollard y tiene un tiempo promedio de ejecución de $O\left(B.\frac{\ln n}{\ln B}\right)$. Básicamente se trata de seleccionar una cota B *suave*. Se dice que un entero n es *suave* con respecto una cota B, si todos sus factores primos son menores o iguales a B. El algoritmo es como sigue:

1) Seleccionar la cota B.
2) a=2
3) Desde que j=2 hasta B hacer
$a = a^j \bmod n$
4) $d = \text{mcd}(a-1, n)$
5) Si 1<d<n entonces d es un factor de n
en caso contrario no se ha encontrado ningún factor de n.

Métodos de factorización modernos.

Los trabajos sobre factorización han sido continuos en los últimos años acrecentados por su posible aplicación al criptoanálisis. Los métodos que se han demostrado más útiles son la criba cuadrática, la criba de campos numéricos y el método de las curvas elípticas. Una explicación detallada de los mismos está fuera del ámbito del siguiente trabajo, una excelente referencia para el tema es [RIE94]. El método de las curvas elípticas da

mejores resultados con números de muy diferente tamaño. La criba cuadrática es el algoritmo más rápido conocido para factorizar números de hasta 150 dígitos, existen varias versiones del algoritmo las más conocidas son la criba cuadrática polinomial múltiple y sus variaciones. La criba de campos numéricos es el más reciente de los tres y su tiempo de ejecución asintótico es el mejor de los tres, sin embargo, para números menores de entre 110 y 135 dígitos es más rápido la criba cuadrática polinominal múltiple.

Los tiempos de ejecución asintóticos de los tres algoritmos obtenidos de [STI95] son:

Criba cuadrática	$O(e^{(1+O(1))\sqrt{\ln n \ln\ln n}})$
Curvas elípticas	$O(e^{(1+O(1))\sqrt{2\ln p \ln\ln p}})$
Criba de campos numéricos	$O\left(e^{(1,92+O(1))(\ln n)^{1/3}(\ln\ln n)^{2/3}}\right)$

El mayor número factorizado hasta el momento es el noveno número de Fermat $F_9 = 2^{2^9}+1 = 2^{512}+1$ de 155 dígitos que fue factorizado utilizando el método de la criba cuadrática con 700 ordenadores en paralelo y un supercomputador en las etapas finales del proceso. En total cuatro meses de trabajo [LEN93].

CONTINUARÁ