Divisibilidad en semigrupos y anillos.

Aladar Peter Santha

Résumée:

Dans ce travaille on expose: des connaissances élémentaires sur les demie-groupes, groupes, les anneaux, la constructions de l'anneau des polynômes d'un seul variable, la théorie de la divisibilité dans le demi-groupe des nombres naturelles et dans les anneaux intègres (en particulier dans Z, Q[X], R[X], Z[i], Z[i][X]), les fonctions de Visual-Basic nécessaires pour les calcules algébriques et pour déterminer le plus grand diviseur commun et le plus petit multiple commun dans les structures algébrique énumérés avant. Finalement, comme une application de la plus grand commun diviseur dans l'anneau Q[X] on expose aussi le calcule des zéros réels des polynômes avec des coefficients dans Z[i] ou Z [ε] ($\varepsilon \neq 0, \varepsilon^2 = 0$).

==========

§.1. Nociones preliminares.

Una ley de composición interna (operación) en el conjunto no vacío **G** es una aplicación del tipo $f : G \times G \to G$. Si D_f es el dominio de definición de f y $D_f = G \times G$ la operación es posible siempre. En vez de $f(x, y)$ se suelen utilizar las notaciones más diversas, como

$$x + y,\ xy,\ x \cdot y,\ x * y,\ x \perp y,\ x \circ y,\ x \oplus y, x \otimes y, \ldots$$

Definición 1.1: Un grupoide es un conjunto no vacío **G**, donde se ha definido una ley de composición (operación) interna.

Para indicar un grupoide, se puede utilizar la notación **(G,*)**, donde **G** es el conjunto de base y * simboliza la ley de composición. Sin embargo, cuando no hay duda sobre la ley de composición, se puede hablar sencillamente del grupoide **G**.

El grupoide **(G,*)** es conmutativo o asociativo si para cualquier $a, b, c \in G$

$$a * b = b * a \tag{1.1}$$

, ó

$$(\mathrm{a} * \mathrm{b}) * \mathrm{c} = \mathrm{a} * (\mathrm{b} * \mathrm{c}), \tag{1.2}$$

, respectivamente.

En el grupoide **(G,*)**, $e \in G$ es un elemento neutro a la derecha si para cualquier $a \in G$**,**

$$a * e = a \tag{1.3}$$

De manera análoga, $e \in G$ es un elemento neutro a la izquierda si, para cualquier $a \in G$,

$$e * a = a \tag{1.4}$$

Luego, $e \in G$ es un elemento neutro (bilateral) del grupoide si e es tanto un elemento neutro a la derecha como a la izquierda. En este caso, para cualquier $a \in G$

$$a * e = e * a = a \tag{1.5}$$

Un grupoide no puede tener más de un elemento neutro, puesto que si existieran dos, e y e', entonces de

$$e * e' = e \quad y \quad e * e' = e'$$

, resultaría que $e = e'$.

Definición 1.2: Un semigrupo es un grupoide asociativo.

En un semigrupo **(G,*)** con el elemento neutro e, $a' \in G$ es un elemento simétrico de $a \in G$ si

$$a * a' = a' * a = e \tag{1.6}$$

En un semigrupo con elemento neutro, un elemento no puede tener más de un elemento simétrico. En efecto, si a' y a'' fueran dos elementos simétricos del elemento a, entonces

$$a'' = a'' * e = a'' * (a * a') = (a'' * a) * a' = e * a' = a'$$

Si la ley de composición del semigrupo se llama adición, su elemento neutro será designado con el símbolo 0, el elemento simétrico de a se llamará elemento opuesto de a y será designado por $-a$. Cuando la ley de composición de del semigrupo se llama multiplicación, el elemento neutro será designado por e, el

elemento simétrico de a se llamará el elemento inverso de a y será designado por a^{-1}.

Definición 1.3: Un grupo es un semigrupo con elemento neutro, donde todos los elementos tienen elemento simétrico.

Si el semigrupo en cuestión es conmutativo, se dice que el grupo es conmutativo (abeliano).

Teorema 1.1.: Si $(G,\cdot)$ es un semigrupo con el elemento neutro e y los elementos $a,b \in G$ son invertibles, entonces los elementos a^{-1} y $a \cdot b$ son también invertibles y

$$\left(a^{-1}\right)^{-1} = a \tag{1.7}$$

$$(a \cdot b)^{-1} = b^{-1} \cdot a^{-1} \tag{1.8}$$

En efecto, si el elemento a es invertible, entonces

$$a \cdot a^{-1} = a^{-1} \cdot a = e$$

, de donde resulta que

$$a^{-1} \cdot a = a \cdot a^{-1} = e$$

, y así a^{-1} es el inverso de a. Para demostrar la segunda parte del teorema, se observa que

$$(\mathrm{a} \cdot \mathrm{b}) \cdot \left(b^{-1} \cdot a^{-1}\right) = a \cdot \left(b \cdot b^{-1}\right) \cdot a^{-1} = a \cdot e \cdot a^{-1} = (a \cdot e) \cdot a^{-1} = a \cdot a^{-1} = e$$

$$\left(b^{-1} \cdot a^{-1}\right) \cdot (a \cdot b) = b^{-1} \cdot \left(a^{-1} \cdot a\right) \cdot b = b^{-1} \cdot e \cdot b = b^{-1} \cdot (e \cdot b) = b^{-1} \cdot b = e$$

, de donde resulta que $b^{-1} \cdot a^{-1}$ es el elemento inverso de $a \cdot b$.

En un grupo aditivo, las igualdades (1.7) y (1.8) se transforman en:

$$-(-a) = a \qquad \text{y} \qquad -(a+b) = (-a) + (-b)$$

, respectivamente.

Teorema 1.2: Si G es un semigrupo con elemento neutro y U es el conjunto de todos los elementos invertibles de G, entonces **U** es un grupo respecto a la ley de composición de **G**.

Demostración: Dado que $e \in U$, **U** no es el conjunto vacío. Por otra parte, según el teorema 1.1, **U** es una parte estable de **G**, es decir, la composición de dos elementos de U pertenece a **U**. Finalmente, si $a \in U$ entonces, según el teorema 1.1, $a^{-1} \in U$ y así **U** será un grupo.

Definición 1.4: Un anillo es un conjunto **A** en el cual se han definido dos leyes de composición internas, llamadas, de manera convencional, adición y multiplicación, y que cumplen las condiciones:

1) **(A,+)** es un grupo abeliano.
2) Cualesquiera que sean $a,b,c \in A$,

$$a \cdot (b+c) = a \cdot b + a \cdot c \tag{1.9}$$

$$(a+b) \cdot c = a \cdot c + b \cdot c \tag{1.10}$$

Si la multiplicación es asociativa (conmutativa), el anillo $(A,+,\cdot)$ es asociativo (conmutativo). Las leyes (1.9) y (1.10) se llaman las leyes de distributividad de la multiplicación respecto a la adición, a la izquierda y a la derecha, respectivamente.

El elemento neutro del grupo **(A,+)** será designado con 0 y se llamará el elemento nulo (cero) del anillo.

El grupoide multiplicativo $(A,\cdot)$ del anillo $(A,+,\cdot)$ puede tener también un elemento neutro y, cuando esto es así, será designado con 1, diciendo que es el elemento neutro del anillo.

Si no hay ninguna duda sobre las operaciones, el anillo $(A,+,\cdot)$ podría ser designado simplemente por **A**.

Poniendo $b = c = 0$ en la igualdad (1.9), se obtiene que

$$a \cdot 0 = a \cdot 0 + a \cdot 0,$$

, de donde, sumando $-a \cdot 0$ en las dos partes de la igualdad, resulta que $a \cdot 0 = 0$, cualquiera que sea $a \in A$.

De manera análoga, de (1.10) se obtiene que $0 \cdot a = 0$, cualquiera que sea $a \in A$. Así pues, cualquiera que sea $a \in A$,

$$a \cdot 0 = 0 \cdot a = 0 \tag{1.11}$$

Si $a \neq 0$ y $b \neq 0$ pero $a \cdot b = 0$, se dice que a y b son divisores de cero.

Definición 1.5: El anillo $(A,+,\cdot)$ es un anillo íntegro si es asociativo y conmutativo, tiene elemento neutro y no tiene divisores de cero.

Ejemplo 1.1: Sea **F** el conjunto de todas las funciones reales. Si $f,g \in F$**,** las funciones $f \pm g$ y $f \cdot g$ se definen por

$$(f \pm g)(x) = f(x) \pm g(x) \quad \text{y} \quad (f \cdot g)(x) = f(x)g(x)$$

, respectivamente. Es fácil de verificar que $(F,+,\cdot)$ es un anillo asociativo y conmutativo, con el elemento neutro θ y con el elemento unidad u, definidas por

$$\forall x \in R, \quad \theta(x) = 0 \quad y \quad u(x) = 1$$

, respectivamente. Este anillo no es íntegro, puesto que si las funciones f y g se definen por

$$f(x) = \begin{cases} 1 & si \;\; x \le 2 \\ 0 & si \;\; x > 2 \end{cases} \quad \text{y} \quad g(x) = \begin{cases} 0 & si \;\; x \le 2 \\ 3 & si \;\; x > 2 \end{cases}$$

, respectivamente, entonces $f \ne \theta, g \ne \theta \;\; y \;\; f \cdot g = \theta$.

Teorema 1.3: Si $(A,+,\cdot)$ es un anillo y $a,b \in A$, entonces

$$a \cdot (-b) = (-a) \cdot b = -a \cdot b \tag{1.12}$$

$$(-a) \cdot (-b) = a \cdot b \tag{1.13}$$

$$(-1) \cdot a = -a \tag{1.14}$$

Demostración: Obviamente

$$a \cdot [b + (-b)] = a \cdot 0 = 0,$$

, de donde, según (1.9),

$$a \cdot b + a \cdot (-b) = 0$$

y así, $a \cdot (-b))$ es el elemento opuesto de $a \cdot b$

De manera similar, de

$$[a + (-a)] \cdot b = 0$$

, y de (2.1.10), resultará que

$$(-a) \cdot b = -a \cdot b \tag{1.15}$$

Por fin,

$$(-a) \cdot (-b) = -[a \cdot (-b)] = -(-a \cdot b) = a \cdot b$$

, y poniendo $a = 1$ en (1.15), se obtiene la igualdad (1.14).

Definición 1.6: Si $(A,+,\cdot)$ es un anillo y $a,b \in A$, entonces

$$a - b = a + (-b)$$

La ley de composición así definida se llama sustracción.

Teorema 1.4: La multiplicación es distributiva respecto a la sustracción, es decir, cualesquiera que sean $a,b,c \in A$

$$a \cdot (b - c) = a \cdot b - a \cdot c \tag{1.16}$$

$$(a - b) \cdot c = a \cdot c - b \cdot c \tag{1.17}$$

En efecto,

$$a \cdot (b - c) = a \cdot [b + (-c)] = a \cdot b + a \cdot (-c) = a \cdot b + (-a \cdot c) = a \cdot b - a \cdot c$$

La demostración de (1.17) es análoga.

Teorema 1.5 (Regla de simplificación):

Si $(A,+,\cdot)$ es un anillo íntegro y $a \in A - \{0\}$, entonces

$$a \cdot b = a \cdot c \quad \Rightarrow \quad b = c$$

En efecto, si $a \cdot b = a \cdot c$ entonces $a \cdot (b - c) = 0$ y, dado que $a \ne 0$, resulta que $b - c = 0$, es decir, b = c.

Definición 1.7: El anillo $(K,+,\cdot)$ es un cuerpo si y solamente si, $(\text{K - }\{0\}, \cdot)$es un grupo.

Teorema 1.6: Un cuerpo conmutativo es un anillo integro.

En efecto, si $a,b \in K$ y $a \cdot b = 0$, entonces de $a \ne 0$ resulta que

$$b = e \cdot b = (a^{-1} \cdot a) \cdot b = a^{-1} \cdot (a \cdot b) = a^{-1} \cdot 0 = 0$$

Definición 1.8: Si $(G,\cdot)$ y $(H,*)$ son semigrupos (grupos) y la aplicación $f : G \to H$ cumple la condición

$$f(a \cdot b) = f(a) \cdot f(b)$$

, cualesquiera que sean $a,b \in G$, f es un homomorfismo de semigrupos (de grupos).Si el homomorfismo f es biyectiva, entonces se dice que f es un isomorfismo de semigrupos (de grupos).

Teorema 1.7: Sea $f: G \to H$ un isomorfismo entre los grupos $(G,\cdot)$ y $(H,*)$. Entonces,

a) Siendo e y e' los elementos neutros de $(G,\cdot)$ y $(H,*)$, $f(e)=e'$

b) Si a' es el elemento simétrico de a en $(G,\cdot)$, $f(a')$ es el elemento simétrico , de $f(a)$ en $(H,*)$.

c) Si $(G,\cdot)$ es un grupo conmutativo, $(H,*)$ lo es también.

Demostración: a) Cualquiera que sea $a \in G$,

$$e \cdot a = a \cdot e = a$$
$$f(e \cdot a) = f(a \cdot e) = f(e)$$
$$f(e) * f(a) = f(a) * f(e) = f(e),$$

, y teniendo en cuenta que $f(a)$ es un elemento cualquiera de **H**, resulta que $f(e)$ es un elemento neutro en $(H,*)$. Puesto que en un grupo el elemento neutro es único $f(e)=e'$.

b) Dado que

$$a' \cdot a = a \cdot a' = e,$$

, cualquiera que sea $a \in G$, resulta que

$$f(a') * f(a) = f(a) * f(a') = f(e) = e'.$$

Por tanto, $f(a')$ es un elemento simétrico de $f(a)$ en $(H,*)$ Luego, según la unicidad del elemento simétrico en un grupo $[f(a)]' = f(a')$.

c) Puesto que $(G,\cdot)$ es un grupo conmutativo $a \cdot b = b \cdot a$, cualesquiera que sean $a,b \in G$. Al ser f un isomorfismo, $f(a)$ y $f(b)$ serán dos elementos cualesquiera de $(H,*)$ y así, de

$$f(a) * f(b) = f(b) * f(a)$$

, resulta que **(H,*)** es un grupo conmutativo.

Es fácil de observar que las afirmaciones del teorema anterior quedarán válidas también en un semigrupo.

Teorema 1.8: Si $(G,\cdot)$ es un semigrupo con el elemento neutro e, $(H,*)$ es un grupoide y existe una biyección $f: G \to H$ tal que

$$f(a \cdot b) = f(a) \cdot f(b),$$

, cualesquiera que sean $a,b \in G$, entonces $(H,*)$ es un semigrupo con el elemento neutro $f(e)$.

Demostración: Dado que f es una aplicación biyectiva, cualesquiera que sean $x,y,z \in H$, existirán $a,b,c \in G$, tales que

$$x = f(a),\ y = f(b), z = f(c).$$

Entonces,

$$x*(y*z) = f(a)*[f(b)*f(c)] = f(a)*f(b \cdot c) = f[a \cdot (b \cdot c)] =$$
$$= f[(a \cdot b) \cdot c] = f(a \cdot b) * f(c) = [f(a) * f(b)] * f(c) = (x*y)*z$$

, y así $(H,*)$ es un semigrupo.

Por otra parte, $f(e)$ es un elemento neutro del semigrupo $(H,*)$, puesto que para cualquier $x \in H$,

$$x * f(e) = f(a) * f(e) = f(a \cdot e) = f(a)$$
$$f(e) * x = f(e) * f(a) = f(e \cdot a) = f(a)$$

Luego, $f(e)$ es el único elemento neutro de $(H,*)$ ya que en un semigrupo el elemento neutro es único.

Teorema 1.9: Si $(G,\cdot)$ es un grupo con el elemento neutro e, $(H,*)$ es un grupoide y existe una aplicación biyectiva $f: G \to H$ tal que $f(a \cdot b) = f(a) * f(b)$, cualesquiera que sean $a,b \in G$, entonces $(H,*)$ es un grupo. Si $(G,\cdot)$ es un grupo conmutativo entonces $(H,*)$ lo es también.

Demostración: Según el teorema 1.8, $(H,*)$ es un semigrupo con el elemento neutro $f(e)$. Si $x = f(a)$ $(a \in G)$ es un elemento cualquiera de H y a' es el elemento simétrico de a en $(G,\cdot)$, entonces $f(a')$ es un elemento simétrico de x en el semigrupo $(H,*)$.

En efecto,

$$x * f(a') = f(a) * f(a') = f(a \cdot a') = f(e)$$

$$f(a')*x = f(a')*f(a) = f(a'\cdot a) = f(e)$$

Teniendo en cuenta que en un semigrupo un elemento no puede tener más de un elemento simétrico $f(a')$ será el único elemento simétrico de $x = f(a)$. Así, todos los elementos de H tienen elemento simétrico en el semigrupo $(H,*)$ y, por tanto, $(H,*)$ es un grupo. Si $(G,\cdot)$ es un grupo conmutativo, entonces

$$x*y = f(a)*f(b) = f(a\cdot b) = f(b\cdot a) = f(b)*f(a) = y*x$$

, cualquiera que sean $x, y \in H$ y así, $(H,*)$ es un grupo conmutativo.

Definición 1.9: Si $(A,+,\cdot)$ y $(B,\oplus,*)$ son anillos, una aplicación $f: A \to B$ es un homomorfismo entre ellos si, para cualquier $x, y \in A$,

$$f(x+y) = f(x) \oplus f(y)$$
$$f(x\cdot y) = f(x)*f(y)$$

Si la aplicación f es biyectiva, f es un isomorfismo de anillos.

Observación 1.1: Si 0 y 0' son los elementos neutros en los grupos aditivos de los anillos $(A,+,\cdot)$ y $(B,\oplus,*)$, respectivamente, y f es un isomorfismo entre estos dos anillos, entonces, aplicando el teorema 1.7 para los semigrupos $(A,+)$ y $(B,\oplus)$

$$f(0) = 0' \quad y \quad f(-a) = -f(a),$$

, cualquiera que sea $a \in A$.

De manera análoga, aplicando el teorema 1.7 para los semigrupos multiplicativos $(A,\cdot)$ y $(B,*)$, si existe el elemento neutro e en $(A,\cdot)$ y el elemento a posee el elemento simétrico a', entonces $e' = f(e)$ es el elemento neutro en $(B,*)$ y $f(a')$ es el elemento simétrico de $f(a)$.

Teorema 1.10: Si en el conjunto H se han definido dos leyes de composición internas $\oplus$ y $*$ y, $f: A \to H$ es una aplicación biyectiva entre el anillo $(A,+,\cdot)$ y el conjunto H, tal que

$$f(x+y) = f(x) \oplus f(y) \qquad (1.18)$$
$$f(x\cdot y) = f(x)*f(y), \qquad (1.19)$$

, cualesquiera que sean $x, y \in A$, entonces $(H,\oplus,*)$ es un anillo isomorfo con el anillo $(A,+,\cdot)$.

Si existe el elemento neutro e en el anillo $(A,+,\cdot)$, entonces $f(e)$ es el elemento neutro del anillo $(H,\oplus,*)$.

Demostración: Puesto que f cumple la condición (1.18), aplicando el teorema 1.8 para el grupo aditivo $(A,+)$ y el grupoide $(H,\oplus)$, resulta que $(H,\oplus)$ es un grupo abeliano.

Luego, aplicando el mismo teorema para el semigrupo $(A,\cdot)$ y el grupoide $(H,*)$ y, teniendo en cuenta que f verifica la condición (1.18), resultará que $(H,*)$ es un semigrupo con el elemento neutro $f(e)$ y es conmutativo si lo era $(A,\cdot)$.

Para que $(H,\oplus,*)$ sea un anillo, hay que comprobar que se cumplen las leyes distributivas. En efecto, dado que f es una aplicación biyectiva, para cualquier $x, y, z \in H$, existirán $a, b, c \in A$ tal que $x = f(a)$, $y = f(b)$ y $z = f$. Entonces,

$$\begin{aligned} x*(y\oplus z) &= f(a)*[f(b)\oplus f(c)] = f(a)*f(b+c) = f[a\cdot(b+c)] \\ &= f(a\cdot b + a\cdot c) = f(a\cdot b)\oplus f(a\cdot c) \\ &= f(a)*f(b)\oplus f(a)*f(b) = x*y \oplus x*z \end{aligned}$$

, cualesquiera que sean $x, y, z \in H$. Igual se demuestra la distributividad a la derecha.

Así pues $(H,\oplus,*)$ es un anillo y según las relaciones (1.18) y (1.19) f es un isomorfismo, luego, según la observación 1.1, $f(e)$ es el elemento neutro de este anillo.

Teorema 1.11: Si $(A,+,\cdot)$ es un anillo íntegro y $f: A \to B$ es un ismorfismo entre los anillos $(A,+,\cdot)$ y $(B,\oplus,*)$, entonces $(B,\oplus,*)$ es también un anillo íntegro.

Demostración: Si $x, y \in A$ cumplen la relación $x*y = f(0)$, donde $f(0)$ es el elemento nulo del anillo $(B,\oplus,*)$, entonces

$$f(a)*f(b) = f(0)$$

$$f(a\cdot b)=f(0)$$

, de donde resultará que $a\cdot b=0$, puesto que f es una aplicación biyectiva. Teniendo ahora en cuenta que $(A,+,\cdot)$ es un anillo íntegro, de la última igualdad resulta que $a=0$ ó $b=0$, es decir,

$$f(a)=f(0)\ \text{, ó}\ \ f(b)=f(0).$$

Así, $(B,\oplus,*)$ será también un anillo íntegro.

Teorema 1.12: Si $f:A\to B$ es un isomorfismo entre los anillos $(A,+,\cdot)$ y $(B,\oplus,*)$, entonces $f^{-1}:B\to A$ lo es también.

En efecto, cualesquiera que sean los elementos $x=f(a)$ e $y=f(b)$ de B,

$$f^{-1}(x\oplus y)=f^{-1}[f(a)\oplus f(b)]=f^{-1}[f(a+b)]=a+b=f^{-1}(x)+f^{-1}(y$$
$$f^{-1}(x*y)=f^{-1}[f(a)*f(b)]=f^{-1}[f(a\cdot b)]=a\cdot b=f^{-1}(x)\cdot f^{-1}(y)$$

Si $A(+,\cdot)$ es un anillo íntegro con el elemento neutro e, sea $F=(p,q)/\,p\in A,q\in A-\{0\})$. En F se puede considerar la relación $\approx$, definida de la manera siguiente:

$$(p,q)\approx(p',q')\ \Leftrightarrow\ p\cdot q'=p'q$$

Es fácil verificar que esta relación es de equivalencia. Por tanto dividirá el conjunto F en clases de equivalencia. La fración $\frac{p}{q}$ es el conjunto de todos los elementos de F equivalentes al par (p,q), es decir,

$$\frac{p}{q}=\{(x,y)/(x,y)\in F\ y\,(x,y)\approx(p,q)\}.$$

Obviamente,

$$\frac{p}{q}=\frac{s}{t}\ \Leftrightarrow\ (p,q)\approx(s,t)\ \Leftrightarrow\ p\cdot t=q\cdot s$$

En el conjunto de las fracciones del anillo, designado por $FCr(A)$, se pueden introducir la addición y la multiplicación de la manera siguiente:

$$\frac{a}{b}+\frac{c}{d}=\frac{ad+bc}{bd}\qquad \frac{a}{b}\cdot\frac{c}{d}=\frac{ac}{bd}\qquad (1.20)$$

Estas definiciones son correctas puesto que si

$$\frac{a}{b}=\frac{a'}{b'}\quad y\quad \frac{c}{d}=\frac{c'}{d'}\qquad \Leftrightarrow\quad ab'=ba'\quad y\quad cd'=dc'$$

, y entonces

$$b'd'(ad+bc)=adb'd'+bcb'd'=bdda'+bb'dc'=bd(a`d'+b'c')\ \Leftrightarrow\ \frac{a}{b}+\frac{c}{d}=\frac{a'}{b'}+\frac{c'}{d¡}$$

$$acb'd'=ba'dc'=a'c'bd\quad \Leftrightarrow \frac{a}{b}\cdot\frac{c}{d}=\frac{a'}{b'}\cdot\frac{c'}{d'}$$

Teorema 1.13 (Regla de la simplificación ó amplificación): Cualquiera que sea $x\in A-\{0\}$, $\frac{px}{qx}=\frac{p}{q}$.

En efecto

$$\frac{px}{qx}=\frac{p}{q}\ \Leftrightarrow\ (px,qx)\approx(p,q)\ \Leftrightarrow\ pqx=qxp.$$

Observación 1.2: Si dos fracciones tienen el mismo denominador, para sumarlas se conserva se suman los numeradores y se conserva el denominador común. En efecto, según el teorema 1.13,

$$\frac{a}{c}+\frac{b}{c}=\frac{ac+bc}{c^2}=\frac{(a+b)c}{cc}=\frac{a+b}{c}$$

Teorema 1.14: Si $A(+,\cdot)$ es un anillo íntegro con el elemento neutro e, $CFr(A)$ es un cuerpo conmutativo con repecto la addición y multiplicación definidas en (1.20) y A será un subanillo en el cuerpo de sus fracciones.

En efecto, la adición y la multiplicación son comutativas puesto que

$$\frac{a}{b}+\frac{c}{d}=\frac{ad+bc}{bd}=\frac{c}{d}+\frac{a}{b}\qquad \frac{a}{b}\cdot\frac{c}{d}=\frac{ac}{bd}=\frac{c}{d}\cdot\frac{a}{b}$$

De las igualdades siguientes resulta que son también asociativas:

$$\left(\frac{a}{b}+\frac{c}{d}\right)+\frac{u}{v}=\frac{ad+bc}{bd}+\frac{u}{v}=\frac{adv+bcv+bdu}{bdv}$$

$$\frac{a}{b}+\left(\frac{c}{d}+\frac{u}{v}\right)=\frac{a}{b}+\frac{cv+du}{dv}=\frac{adv+bcv+bdu}{adv}$$

$$\left(\frac{a}{b}\cdot\frac{c}{d}\right)\cdot\frac{u}{v}=\frac{ac}{bd}\cdot\frac{u}{v}=\frac{(ac)u}{(bd)v}=\frac{a(cu)}{b(dv)}=\frac{a}{b}\cdot\left(\frac{c}{d}\cdot\frac{u}{v}\right)$$

La multiplicación es distributiva respecto la adición: Utilizando el teorema 1.13.

$$\frac{u}{v}\cdot\left(\frac{a}{b}+\frac{c}{d}\right)=\frac{u}{v}\cdot\frac{ad+bc}{bd}=\frac{uad+ubc}{vbd}=\frac{uad}{vbd}+\frac{ubc}{vbd}=\frac{ua}{vb}+\frac{uc}{vd}=\frac{u}{v}\cdot\frac{a}{b}+\frac{u}{v}\cdot\frac{c}{d}$$

Admitiendo que para $a\in A,\ \frac{a}{e}=a$ resultará que $A\subset CFr(A)$. Así para $a\neq 0$ tenemos también

$$0=\frac{0}{e}=\frac{0}{a}\text{, y } \frac{a}{a}=\frac{e}{e}=e$$

Finalmente, $CFr(A)$ es un cuerpo puesto que:

$$\frac{p}{q}+0=\frac{p}{q}+\frac{0}{q}=\frac{p+0}{q}=\frac{p}{q}\ ,\ \frac{p}{q}\cdot e=\frac{p}{q}\cdot\frac{e}{e}=\frac{p}{q}\quad (a\neq 0)$$

$$\frac{p}{q}+\frac{-p}{q}=\frac{-pq+pq}{pq}=\frac{0}{pq}=0\quad\Rightarrow\quad -\frac{p}{q}=\frac{-p}{q}$$

$$\left(\frac{p}{q}\right)^{-1}=\frac{q}{p}\text{, puesto que } \frac{p}{q}\cdot\frac{q}{p}=\frac{pq}{qp}=e\quad\Rightarrow\quad\left(\frac{p}{q}p\right)^{-1}=q$$

§.2. Construcción del anillo de los polinomios.

Si A es un anillo conmutativo sea $A(S)$ el conjunto de todas las sucesiones $(u_k)_{k\in N}$, donde $u_k=0$ solo para un número finito de $k\in N$.

Si $\alpha,\beta\in A(S)$, es decir, $\alpha=(u_k)_{k\in N}$ y $\beta=(v_k)_{k\in N}$, entonces las sucesiones

$$\alpha+\beta=(u_k+v_k)_{k\in N}\quad\text{y}\quad\alpha\cdot\beta=(w_k)_{k\in N}$$

, donde $w_k=\sum_{\lambda+\mu=k}u_\lambda v_\mu$, pertenecen al conjunto $A(S)$.

En efecto, si $\alpha,\beta\in A(S)$, entonces existen $k_\alpha,k_\beta\in N$ tales que

$$u_{k_\alpha}\neq 0\quad,\quad u_{k_\beta}\neq 0$$

$$k>k_\alpha\Rightarrow u_k=0\quad y\quad k>k_\beta\Rightarrow v_k=0$$

Si $k^*=\max(k_\alpha,k_\beta)$, entonces

$$k>k^*\quad\Rightarrow\quad u_k=0\ \ y\ \ v_k=0$$

, y así $\alpha+\beta\in A(S)$.

Por otra parte, si

$$k=\lambda+\mu>k_\alpha+k_\beta$$

, entonces

$$\lambda>k_\alpha\ \ ó\ \ \mu>k_\beta\,,$$

, es decir

$$u_\lambda=0\ \ ó\ \ v_\mu=0\,.$$

Así pues, si $k>k_\alpha+k_\beta$, entonces $w_k=0$ y, por tanto $\alpha\cdot\beta\in A(S)$.

Por consiguiente, en $A(S)$ se pueden considerar las leyes de composición:

$$(\alpha,\beta)\to\alpha+\beta\quad\text{y}\quad(\alpha,\beta)\to\alpha\cdot\beta$$

Teorema 2.1: $(A(S),+,\cdot)$ es un anillo.

Demostración: Si $\alpha,\beta,\gamma\in A(S)$ y

$$\alpha = (a_k)_{k\in N} \quad , \quad \beta = (b_k)_{k\in N} \quad \text{y} \quad \gamma = (c_k)_{k\in N} \quad ,$$

, entonces, puesto que la adición en el anillo **A** es conmutativa y asociativa,

$$(\alpha+\beta)+\gamma = ((a_k+b_k)+c_k)_{k\in N} = (a_k+(b_k+c_k))_{k\in N} = \alpha+(\beta+\gamma)$$

, y

$$\alpha+\beta = (a_k+b_k)_{k\in N} = (b_k+a_k)_{k\in N} = \beta+\alpha$$

Así (A(S),+) es un semigrupo conmutativo. Obviamente,

$$O = (a_k)_{k\in N} \quad ,$$

, donde $a_k = 0$ para cualquier $k \in N$, es el elemento neutro de este semigrupo y

$$-\alpha = (-a_k)_{k\in N}$$

, es el opuesto de α. Por tanto (A(S),+) es un grupo conmutativo con el elemento neutro 0.

La multiplicación de $A(S)$ es asociativa puesto que si

$$\alpha\cdot\beta = (p_k)_{k\in N} \quad ; \quad \beta\cdot\gamma = (p_k^{'})_{k\in N}$$

$$(\alpha\cdot\beta)\cdot\gamma = (q_j)_{j\in N} \quad ; \quad \alpha\cdot(\beta\cdot\gamma) = (q_j^{'})_{j\in N}$$

, entonces

$$p_k = \sum_{\lambda+\mu=k} a_\lambda b_\mu \qquad p_k^{'} = \sum_{\mu+\lambda=k} b_\mu c_\lambda$$

$$q_j = \sum_{k+\lambda=j} p_k c_\lambda = \sum_{k+\lambda=j}\left\{\sum_{\nu+\mu=k} a_\nu b_\mu\right\} = \sum_{\nu+\mu+\lambda=j} a_\nu b_\mu c\lambda$$

$$q_j^{'} = \sum_{\nu+k=j} a_\nu p_k^{'} = \sum_{\mu+k=j} a_\nu\left\{\sum_{\nu+\lambda=k} b_\nu c_\lambda\right\} = \sum_{\nu+\mu+\lambda=j} a_\nu b_\mu c_\lambda$$

, y así, para cualquier $j \in N$ $q_j = q_j^{'}$, es decir, cualesquiera que sean $\alpha,\beta,\gamma \in A(S)$,

$$(\alpha\cdot\beta)\cdot\gamma = \alpha\cdot(\beta\cdot\gamma)$$

La multiplicación de $A(S)$ es también conmutativa, puesto que si

$$\alpha\cdot\beta = (p_k)_{k\in N} \quad \text{y} \quad \beta\cdot\alpha = (p_k^{'})_{k\in N}$$

, entonces

$$p_k = \sum_{\lambda+\mu=k} a_\lambda b_\mu = \sum_{\mu+\lambda=k} b_\mu a_\lambda$$

, cualquiera que sea $k \in N$ y así,

$$\alpha\cdot\beta = \beta\cdot\alpha$$

, cualesquiera que sean $\alpha,\beta \in A(S)$.

Para demostrar que $A(S)$ es un anillo queda verificar que la multiplicación es distributiva respecto a la adición. Si

$$\alpha\cdot\gamma = (p_k^{'})_{k\in N} \quad , \quad \beta\cdot\gamma = (p_k^{''})_{k\in N} \quad \text{y} \quad (\alpha+\beta)\cdot\gamma = (p_k)_{k\in N}$$

, entonces

$$p_k = \sum_{\lambda+\mu=k}(a_\lambda+b_\lambda)c_\mu = \sum_{\lambda+\mu=k} a_\lambda c_\mu + \sum_{\lambda+\mu=k} b_\lambda c_\mu = p_k^{'} + p_k^{''}$$

, cualquiera que sea $k \in N$ y así

$$(\alpha+\beta)\cdot\gamma = \alpha\cdot\gamma+\beta\cdot\gamma$$

, cualesquiera que sean $\alpha,\beta,\gamma \in A(S)$.

Teorema 2.2: Si e es el elemento neutro del anillo **A**, entonces

$$e^* = (u_k)_{k\in N}$$

, donde $u_0 = e \;\; y \;\; u_k = 0$ para $k \neq 0$, es el elemento neutro del anillo $A(S)$.

En efecto, si $\alpha = (a_k)_{k\in N}$ es un elemento cualquiera de A(S), entonces

$$e^* \cdot \alpha = (b_k)_{k\in N}$$

, donde

$$b_k = \sum_{\lambda+\mu=k} u_\lambda a_\mu = a_k$$

, cualquiera que sea $k \in N$ y así $e^* \cdot \alpha = \alpha$.

En los siguientes supondremos que el anillo **A** posee un elemento neutro e y sea $E = (e_k)_{k\in N}$, donde

$$e_k = \begin{cases} 0 & si \;\; k \neq 1 \\ e & si \;\; k = 1 \end{cases}$$

Lema 2.1: S $\alpha = (a_k)_{k\in N} \in A(S)$ y $\beta = (b_k)_{k\in N} = \alpha \cdot E$, entonces

$$b_k = \begin{cases} 0 & si \quad k = 0 \\ a_{k-1} & si \quad k \neq 0 \end{cases}$$

En efecto,

$$b_0 = \sum_{\lambda+\mu=k} a_\lambda e_\mu = a_0 e_0 = 0$$

, y si $k \geq 1$, entonces $b_k = \sum_{\lambda+\mu=k} a_\lambda e_\mu = a_k e_0 + a_{k-1} e_1 + \cdots + a_0 e_k$

Luego, puesto que $e_0 = 0$, $e_1 = e$ y $e_i = 0$ para $i > 1$, resulta que $b_k = a_{k-1}$.

Teorema 2.3: Si $E^0 = (e,0,0,\ldots)$ y $E^1 = (0,e,0,0,\ldots)$, entonces

$$E^n = (a_{k,n})_{k\in N} \qquad n > 1 \quad \text{y} \quad a_{k,n} = \begin{cases} e & si \;\; k = n+1 \\ 0 & si \;\; k \neq n+1 \end{cases} \tag{2.1}$$

Demostración: En efecto,

$$E^2 = E \cdot E = (a_{k,2})_{k\in N} \;,$$

, donde según el lema 2.1, $a_{2,2} = e$ y para $k \neq 2$ tenemos $a_{k,2} = 0$. Así, la igualdad (2.1) se verifica para $n = 2$. Suponiendo que el teorema se verifica para n, demostraremos que será verdadera también para n+1. En efecto,

$$E^{n+1} = E^n \cdot E = (a_{k,n})_{k\in N} \cdot E$$

Según el lema 2.1,

$$a_{n+2,n+1} = a_{n+1,n} = e \;.$$

Si $k \neq n+2$, entonces $k-1 \neq n+1$ y $a_{k,n+1} = a_{k-1,n} = 0.$

Así la relación (2.1) es verdadera también cuando n se sustituye por $n+1$; por tanto el teorema queda demostrado.

Observación 2.1: Cualquiera que sea n,

$$E^n = \left(\underbrace{0,0,\ldots,0}_{n-ceros}, e, 0,0,\ldots \right) \tag{2.2}$$

Lema 2.2: Si $\alpha = (a,0,0,\ldots)$, entonces

$$E^n = \left(\underbrace{0,0,\ldots,0}_{n-ceros}, a, 0,0,\ldots \right) \tag{2.3}$$

En efecto, sea $\alpha = (a_k)_{k\in N}$, donde $a_0 = a$ y $a_k = 0$ para $k > 0$. Entonces,

$$\alpha \cdot E = (b_k)_{k\in N} \;,$$

, donde

$$b_k = \sum_{\lambda+\mu=k} a_\lambda a_{\mu,n} = a_0 a_{k,n} + a_1 a_{k-1,n} + \cdots + a_k a_{0,n} = a\, a_{k,n}$$

, puesto que $a_k = 0$ para $k > 0$. Así, según el teorema 2.1.9, $b_{n+1} = a$ y $b_k = 0$ para $k \neq n+1$.

Teorema 2.4: Si $\alpha = (a_i)_{i\in N} \in A(S)$, entonces existirá $n \in N$ tal que

$$\alpha = (a_0,0,\ldots)+(a_1,0,\ldots)\cdot E+\cdots+(a_n,0,\ldots)\cdot E^n \qquad (2.4)$$

y si $\alpha \neq (0,0,\ldots)$, esta escritura de α es única .

En efecto, si $\alpha = (0,0,\ldots)$, entonces la relación (2.2.4) se verifica o bien para $n=0$ y $a_0=0$, o bien para n cualquiera y $a_0 = a_1 = \cdots = a_n = 0$. En el caso contrario, teniendo en cuenta que $a_k = 0$ solamente para un número finito de valores de k, existirá $n \in N$ tal que $a_n \neq 0$ y

$$\alpha = (a_0,a_1,\ldots,a_n,0,0,\ldots) = (a_0,0,\ldots)+(0,a_1,0,\ldots)+\cdots+(0,0,\ldots,a_n,0,\ldots) \qquad (2.5)$$

Así, según el lema 2.2.2, α tendrá la forma siguiente:

$$\alpha = (a_0,0,\ldots)+(a_1,0,\ldots)\cdot E+\cdots+(a_n,0,\ldots)\cdot E^n$$

Si para $\alpha \neq (0,0,\ldots)$ tendríamos también

$$\alpha = (b_0,0,\ldots)+(b_1,0,\ldots)\cdot E+\cdots+(b_n,0,\ldots)\cdot E^n$$

, entonces resultaría que

$$\alpha = (b_0,0,\ldots)+(0,b_1,0,\ldots)+\cdots+(0,0,\ldots,b_n,0,\ldots) = (b_0,b_1,\ldots,b_n,0,\ldots) \quad (2.6)$$

, y la comparación entre (2.5) y (2.6) daría que $m=n$ y $a_k = b_k \quad (k=1,2,\ldots,n)$.

Sea ahora $A[X]$ el conjunto de todas las expresiones de la forma:

$$a_0 + a_1X + \cdots + a_nX^n \qquad (2.7)$$

, donde $a_k \in A,\ n \in N$ y

$$a_n = 0 \quad \Rightarrow \quad a_0 = a_1 = \cdots = a_{n-1} = 0$$

Se dice que la expresión (2.7) es un polinomio en la indeterminada X y con los coeficientes a_k, pertenecientes al anillo A. Si $a_n \neq 0$, se dice que el polinomio es de grado n. Cuando todos los coeficientes son nulos se trata del polinomio nulo, notado con $\theta(X)$ ó 0, cuyo grado es indeterminado.

Definición 2.1: Dos polinomios son iguales si los dos son nulos, ó bien si tienen el mismo grado y los coeficientes correspondientes a las mismas potencias de X son iguales.

Si

$$P(X) = a_0 + a_1X + \cdots + a_nX^n \qquad (2.8)$$

$$Q(X) = b_0 + b_1X + \cdots + b_mX^m \qquad (2.9)$$

, son dos polinomios de grado n y m, respectivamente $(n \leq m)$, entonces en $A[X]$ se puede definir una adición y una multiplicación de la manera siguiente:

$$P(X)+Q(X) = c_0 + c_1X + \cdots + c_mX^m$$

$$P(X)\cdot Q(X) = d_0 + d_1X + \cdots + d_pX^p$$

, donde $p = mn$,

$$c_i = \begin{cases} a_i + b_i & si \quad i \leq n \\ b_i & si \quad i > n \end{cases} \quad \text{y} \quad d_i = \sum_{j+k=i} a_j b_k$$

Sea ahora

$$f : A(S) \to A[X] \qquad (2.10)$$

, la aplicación definida por:

$$f(\alpha) = a_0 + a_1X + \cdots + a_nX^n$$

, donde

$$\alpha = (a_0,0,\ldots)+(a_1,0,\ldots)\cdot E+\cdots+(a_n,0,\ldots)\cdot E^n$$

Evidentemente, f es suprayectiva y, para demostrar que f es inyectiva, sea

$$\beta = (b_0,0,\ldots)+(b_1,0,\ldots)\cdot E+\cdots+(b_m,0,\ldots)\cdot E^m$$

, tal que

$$f(\alpha) = f(\beta) \qquad (2.11)$$

Si $f(\alpha)$ y $f(\beta)$ son nulos, según la definición 2.2.1,

$$a_0 = a_1 = \cdots a_n = b_0 = b_1 = \cdots b_m = 0$$

, así $\alpha = \beta$.
Si $f(\alpha)$ y $f(\beta)$ no son nulos, entonces según la definición (2.1), $n = m$ y $a_k = b_k$, cualquiera que sea $k \in \{1,2,\ldots n\}$. Por tanto $\alpha = \beta$.
Así pues,

$$f(\alpha) = f(\beta) \quad \Rightarrow \quad \alpha = \beta$$

, y por consiguiente, f es inyectiva. Al ser f inyectiva y suprayectiva a la vez, resulta que f es biyectiva. Por otra parte, si

$$\alpha = (a_0, a_1, \ldots, a_n, 0, \ldots) \quad \text{y} \quad \beta = (b_0, b_1, \ldots, b_m, 0, \ldots)$$

, donde $n \le m$, entonces

$$f(\alpha) = P(X) \quad \text{y} \quad f(\beta) = Q(X)$$

, donde $P(X)$ y $Q(X)$ son los polinomios definidos en (2.2.8) y (2.2.9), respectivamente.
Puesto que

$$f(\alpha) + f(\beta) = P(X) + Q(X) = f(\alpha + \beta)$$
$$f(\alpha) \cdot (\beta) = P(X) \cdot Q(X) = f(\alpha \cdot \beta) \ ,$$

, según el teorema 1.10 $A[X]$ es un anillo y f es un isomorfismo entre los anillos $A(S)$ y $A[X]$.
Observación 2.1: Los anillos $A[X]$ y $A[Y]$ son isomorfos puesto que los dos son isomorfos con el anillo $A(S)$.
Definición 2.2: Un polinomio $M[X]$ se llama monomio si $M(X) = a$ ó $M(X) = aX^n$
, donde $a \in A$ y $n \in N - \{0\}$.
Los monomios aX^n y bX^m son semejantes si y solamente si $n = m$. Obviamente,

$$(aX^n) \cdot (bX^m) = abX^{n+m} \quad \text{y} \quad aX^n + bX^n = (a+b)X^n$$

Observación 2.2: Cualquier polinomio, que no es un monomio, es una suma de monomios. Teniendo en cuenta que la multiplicación es distributiva respecto a la adición en el anillo $A[X]$, resulta que el producto de dos polinomios se puede calcular sumando a todos los monomios, obtenidos multiplicando todos los monomios del primer polinomio con todos los monomios del segundo polinomio.

Observación 2.3: Conviene saber que la suma de dos polinomios de grado n podría no ser un polinomio de grado n.
Por ejemplo, si $n = 3$, $P(X) = 2 - 3X + 5X^2 - 2X^3$ y $Q(X) = -5 + 2X^3$, entonces

$$P(X) + Q(X) = -3 - 3X + 5X^2$$

, que es un polinomio de grado 2. En general,

$$grado(P(X) + Q(X)) \le \max(grado(P(X)), grado(Q(X))),$$

, donde la igualdad tiene lugar cuando los polinomios son de grados distintos, o bien si son de mismo grado y la suma de los coeficientes de los monomios de más alto grado no es cero.

Observación 2.4: Si **A** es un anillo íntegro,

$$grado[P(X) \cdot Q(X)] = grado[P(X)] + grado[Q(X)].$$

En efecto,

$$a_n \ne 0 \ \ y \ \ b_m \ne 0 \quad \Rightarrow \quad a_n b_m \ne 0$$

, y así

$$grado[P(X) \cdot Q(X)] = n + m = grado[P(X)] + grado[Q)(X)].$$

Teorema 2.5: Si **A** es un anillo íntegro, entonces $A[X]$ es también un anillo íntegro.
Demostración: Si $\theta(X)$ es el polinomio nulo, hay que demostrar que

$$P(X) \ne \theta(X) \ \ y \ \ Q(X) \ne \theta(X) \quad \Rightarrow \quad P(X) \cdot Q(X) \ne \theta(X)$$

Puesto que $P(X)$ y $Q(X)$ no son polinomios nulos, se puede suponer que

$$P(X) = a_0 + a_1 X + \cdots + a_n X^n \quad \text{y} \quad Q(X) = b_0 + b_1 X + \cdots + b_m X^m$$

, donde $a_n \neq 0$ y $b_m \neq 0$. Dado que

$$P(X).Q(X) = c_0 + c_1 X + \cdots + c_{n+m} X^{n+m},$$

, donde $c_{n+m} = a_n b_m \neq 0$, resulta que $P(X) \cdot Q(X) \neq \theta(X)$.

Observación 2.5: Puesto que Z, Q, R, C son anillos íntegros, resulta que $Z[X]$, $Q[X]$ $R[X]$ y $C[X]$ son también anillos íntegros.

Observación 2.6: Trabajando con ordenadores, los coeficientes del polinomio

$$P(X) = a_0 + a_1 X + \cdots + a_n X^n = a_n X^n + a_{n-1} X^{n-1} + \cdots + a_1 X + a_0$$

, se introducen siempre en una matriz unidimensional (designado, por ejemplo, con A) de dimensión n, de la maera siguiente:

$$A(0) = a_n\ , A(1) = a_{n-1}\ , \ldots , A(n-1) = a_1\ , A(n) = a_0$$

Utilizando en un ordenador el Lenguaje Visual-Basic, las operaciones con polinomios se pueden realizar mediante las funciones siguientes:

```
Public Function MultPol(ByRef p1() As Double, ByRef p2() As Double) As Variant
        Dim i As Integer, j As Integer, g1 As Integer, g2 As Integer, p () As Double
        g1 = Ubound (p1()):g2 = Ubound(p2())
        ReDim p(g1 + g2)
        For  i = 0 To g1
                If p1(i) <> 0 Then
                        For j = 0 To g2
                                        If p2(j) <> 0 Then
                                        p(i + j) = p(i + j) + p1(i) * p2(j)
                                        End If
                        Next j
                End If
        Next i
        MultPol = p()
End Function
‘-----------------------------------------------------------------------------------------------------------
Public Function SumPol(ByRef p1() As Double, ByRef p2() As Double) As Variant
        Dim gg As Integer, g1 As Integer, g2 As Integer, gp As Integer, gr As Integer
        Dim pt() As Double, r() As Double
        g1 = UBound(p1()): g2 = UBound(p2())
        If g1 >= g2 Then gp = g1 Else gp = g2
        ReDim pt(gp)
        gg = Abs(gp - g1)
        For i = 0 To g1
                pt(i + gg) = p1(i)
        Next i
        gg = Abs(gp - g2)
         For i = 0 To g2
                pt(i + gg) = pt(i + gg) + p2(i)
        Next i
         'Grado de la suma
        For i = 0 To gp
                If pt(i) <> 0 Then Exit For
        Next i
        If i <= gp Then
                gr = gp - i
                ReDim r(gr)
                For j = 0 To gr
                        r(j) = pt(i + j)
                Next j
        Else
                gr = -1
                cxp = "Polinomio nulo"
        End If
        SumPol = r()
End Function
‘-----------------------------------------------------------------------------------------------------------
Public Function RestPol(ByRef p1() As Double, ByRef p2() As Double) As Variant
        Dim gg As Integer, g1 As Integer, g2 As Integer, gp As Integer, gr As Integer
        Dim pt() As Double, r() As Double
        g1 = UBound(p1()): g2 = UBound(p2())
```

```
        If g1 >= g2 Then gp = g1 Else gp = g2
        ReDim pt(gp)
         gg = Abs(gp - g1)
        For i = 0 To g1
                pt(i + gg) = p1(i)
        Next i
        gg = Abs(gp - g2)
         'Grado de la diferencia
         For i = 0 To gp
                If pt(i) <> 0 Then Exit For
        Next i
        If i <= gp Then
                gr = gp - i
                ReDim r(gr)
                For j = 0 To gr
                        r(j) = pt(i + j)
                Next j
        Else
                gr = 0
                ReDim r(0)
                r(0) = 0
                cxp = "Polinomio nulo"
        End If
        RestPol = r()
End Function
'-----------------------------------------------------------------------------------------------------------
Public Function MultPolNum(ByVal r As Double, ByRef x() As Double) As Variant
        Dim m() As Double, i As Integer, g As Integer
        g = UBound(x())
        ReDim m(g)
        For i = 0 To g
                m(i) = r * x(i)
        Next i
        MultPolNum = m()
End Function
'-----------------------------------------------------------------------------------------------------------------------------
Public FunctionTransfOpPol(ByVal cf As Double, ByRef x() As Double, ByVal ex As Integer) As Variant
        Dim i As Integer, j As Integer, g3 As Integer
        Dim x1() As Double, x2() As Double, p() As Double
        If ex > 1 Then
                x1() = x(): x2() = x()
                For j = 1 To ex - 1
                        x1() = ProductoPol(x1(), x2())
                Next j
                p() = ProductoNumPol(cf, x1())
        Else
                If ex = 0 Then
                         gp = 0
                        ReDim p(0)
                        p(0) = cf
                Else
                        p() = ProductoNumPol(cf, x())
                End If
        End If
        TransfOpPol = p()
End Function
'-------------------------------------------------------------------------------------------------------------
Public Function FormatoPol(ByRef x() As Double) As String
        Dim i As Integer, j As Integer, cd As String, cm As String
        gx = UBound(x())
        For i = 0 To gx
                If x(i) <> 0 Then
                        If i = 0 Then
                                If Abs(x(0)) <> 1 Then
                                        cm = FormatoNumero(x(0))
                                Else
                                        If gx <> 0 Then
                                                If x(0) = -1 Then
                                                        cm = "-"
                                                End If
                                        Else
                                                If x(0) = -1 Then
```

```
                                        cm = Str$(-1)
                                Else
                                        cm = Mid$(Str$(1), 2)
                                End If
                        End If
                End If
                If gx <> 0 Then
                        If gx = 1 Then
                                cm = cm + " X"
                        Else
                                cm = cm + " X^" + Mid$(Str$(gx), 2)
                        End If
                End If
        Else
                If x(i) > 0 Then
                        cd = " + "
                Else
                        cd = " - "
                End If
                If Abs(x(i)) <> 1 Or i = gx Then
                        cd = cd + FormatoNumero(Val(Mid$(Str$(x(i)), 2)))
                End If
                If gx > 1 Then
                        If i < gx - 1 Then
                                cd = cd + " X^"
                                cd = cd + Mid$(Str$(gx - i), 2)
                        Else
                                If i = gx - 1 Then
                                        cd = cd + " X"
                                End If
                        End If
                End If
                cm = cm + cd: cd = ""
        End If
    End If
Next i
FormatoPol = cm
End Function
```

La función *CalculoOperando* evalúa una expresión de la forma $r \cdot [P(X)]^n$. Si $n = 1$ **y** $r \neq 1$ entonces se trata solamente de la multiplicación del polinomio con un número. Si $r = 1$ y $n > 1$ se trata de la potencia de un polinomio. Finalmente, en el caso general $r \neq 1$ *y* $n > 1$ se calcula la potencia del polinomio y el resultado se multiplica por el número.

Ejemplo 2.1: Utilizando la función *TransfOpPd* se obtiene el desarrollo del operando

$$Q(X) = 7 \cdot \left(2X^3 - 5X^2 + 3X - 4\right)^3$$

, que es la siguiente:

$$56X^9 - 420X^8 + 1302X^7 - 2471X^6 + 3633X^5 - 4053X^4 + 3381X^3 - 2436X^2 + 1008X - 448$$

Ejemplo 2.3: Utilizando la función *TransfOpPd* se obtiene el desarrollo del operando

$$S(x) = \left(5x^2 - 3x + 2\right)^4 = 1 \cdot \left(5x^2 - 3x + 2\right)^4$$

, que es la siguiente:

$$R(x) = 625x^{12} - 1500x^{10} + 1000x^9 + 1350x^8 - 1800x^7 + 60x^6 + 1080x^5 - 639x^4 - 56x^3 + 216x^2 - 96x + 16$$

El desarrollo de los operandos de los ejemplos anteriores podría ser útil, por ejemplo, para hallar las funciones primitivas de las funciones $x \to Q(x)$ ó $x \to S(x)$. Si se quiere escribir un programa para operar con polinomios, la introducción de los datos desde el teclado es mejor hacerlo en forma de operandos, aunque para esto hay que introducir coeficientes y exponentes iguales a 1. Después de introducir los operandos, estos se transforman en polinomios con la función *TransfOpPd* y luego las operaciones se hacen con las funciones para operar con polinomios.

§.3. Divisibilidad en un semigrupo conmutativo con elemento neutro , y con la regla de simplificación.

Definición 3.1: En el semigrupo $(G,.)$ es válida la regla de simplificación si

$$a.b = a.c \;\Rightarrow\; a = c$$

, cualesquiera que sean $a, b, c \in G$.

Definición 3.2: $a \in G$ es un divisor de $b \in G$ (b es un múltiplo de a) si existe $c \in G$ tal que $b = a.c$.

Observación 4.1: Los divisores del elemento neutro e del semigrupo G son los elementos invertibles de este semigrupo.

Observación 3.2: Un elemento invertible ε (en particular el elemento neutro e) es divisor de cualquier elemento a del semigrupo G. En efecto, $a = \varepsilon.(\varepsilon^{-1}.a)$.

Observación 3.3: Cualquier elemento a del semigrupo G es divisor (múltiplo) de si mismo puesto que $a = a.e$.

Sean D y M las relaciones definidas en el semigrupo G por

$$bDa \;\Leftrightarrow b\ es\ divisible\ con\ a \;\Leftrightarrow\; a\ es\ divisor\ de\ b$$
$$bMa \;\Leftrightarrow\; b\ es\ múltiplo\ de\ a$$

Obviamente, $D^{-1} = M$ y, según la observación 2.4.3, D es reflexiva. Es también transitiva. En efecto, si $cDb\ y\ bDa$ entonces existirán $q\ y\ q'$ tales que

$$b = a.q \;\; y \;\; c = b.q'$$

, de donde resulta que

$$c = a.(q.q')$$

, y así cDa.

Definición 3.3: Los elementos a y b del semigrupo G están asociados si $a = b.q$ y q es un elemento invertible.

La relación de asociación es una relación reflexiva, simétrica y transitiva y, por tanto, una relación de equivalencia. En efecto,

$$a = a.e \;\; y \;\; e^{-1} = e$$
$$a = b.q \;\; y \;\; \exists q^{-1} \;\Rightarrow\; b = a.q^{-1}$$
$$a = b.q\,,\, b = c.r \;\; y \;\; \exists q^{-1}, r^{-1} \;\Rightarrow\; a = c.(r.q) \;\; y \;\; (r.q)^{-1} = q^{-1}.r^{-1}$$

Teorema 3.1: Si bDa y aDb entonces los elementos $a\ y\ b$ están asociados.

En efecto, de bDa y aDb resulta que $b = a.q$ y $a = b.q'$. Así,

$$a.e = a = b.q' = (a.q).q' = a.(q.q')$$

, de donde, aplicando la regla de simplificación, resulta que

$$e = q.q' \;\Leftrightarrow\; q\ y\ q'\ son\ inversibles.$$

Por tanto, $a\ y\ b$ están asociados.

Teorema 3.2: Si $a\ y\ b$ están asociados entonces bDa y aDb. En efecto, existe un elemento invertible q tal que

$$b = a.q \;\Rightarrow\; a = b.q^{-1}$$

Observación 3.4: Si en el semigrupo G el único elemento invertible es el elemento neutro, entonces la relación D (M) será anti-simétrica y así será una relación de orden parcial en **G**.

Observación 3.5: Si a y b son dos elementos asociados y a es un divisor de c, entonces b es también un divisor de c. En efecto, $c = a.q = (b.u).q = b.(u.q)$ donde u es un elemento invertible.

Definición 3.4: $c \in G$ es un divisor común de $a \in G$ y $b \in G$ si c es tanto un divisor de a como de b.

Definición 3.5: $d \in G$ es un máximo común divisor de $a \in G$ y $b \in G$ si d es un divisor común de estos elementos y cualquier otro divisor común d' de a y b es un divisor de d.

De costumbre, un máximo común divisor de los elementos a y b se nota con (a,b).

Definición 3.6: $c \in G$ es un múltiplo común de $a, b \in G$ si tanto a como b son divisores de c.

Definición 3.7: $c \in G$ es un mínimo común múltiplo de $a,b \in G$ si m es un múltiplo común de estos elementos y cualquier otro múltiplo común de a y b es un divisor de m. Un mínimo común múltiplo de a y b se nota con $[a,b]$.

Teorema 3.3: Si tanto d como d_1 es un máximo común divisor de a y b, entonces d y d_1 están asociados.

En efecto, $d_1 D d$ y $d D d_1$ y así, según el teorema 2.4.2, d y d_1 están asociados.

Teorema 3.4: Si d es un máximo común divisor de a y b, y d_1 está asociado con d, entonces d_1 es también un máximo común divisor de a y b.

Demostración: Puesto que d y d_1 están asociados, existirá un elemento invertible q tal que $d = d_1.q$. Entonces

$$a = d.c_1 = (d_1.q).c_1 = d_1.(q.c_1) = d_1.c_1^{'}$$
$$b = d.c_2 = (d_1.q).c_2 = d_1.(q.c_2) = d_1.c_2^{'}$$

y por tanto, d_1 es un divisor común de a y b. Si d_0 es un otro divisor común de a y b, entonces $d = d_0.q_0$ puesto que d es un máximo común divisor de a y b. Así

$$d_1 = d.q^{-1} = (d_0.q_0).q^{-1} = d_0.(q_0.q^{-1})$$

, y así d_0 es un divisor de d_1. Por tanto, d_1 es un máximo común divisor de a y b.

Teorema 3.5: Si tanto m como m_1 es un mínimo común múltiplo de a y b entonces m y m_1 están asociados.

Teorema 3.6: Si m es un mínimo común múltiplo de a y b y m_1 está asociado con m, entonces m_1 es también un mínimo común múltiplo de a y b.

Las demostraciones de los dos últimos teoremas son análogas a la demostración del teorema 2.4.4.

Observación 3.6: En un semigrupo conmutativo con elemento neutro y donde es válida la regla de simplificación, no es seguro la existencia del máximo común divisor o del mínimo común múltiplo de dos elementos, pero si existen, son determinados hasta un factor invertible. En el semigrupo de los números naturales no nulos no existe otro elemento invertible que el elemento neutro 1 y, por tanto, el máximo común divisor y el mínimo común múltiplo son únicos.

Definición 3.8: Los elementos $a,b \in G$ son primos entre sí, si su máximo común divisor está asociado con el elemento neutro e, es decir, si es un elemento invertible.

Teorema 3.7: Si d es un máximo común divisor de a y b, $a = d.a'$ y $b = d.b'$, entonces a' y b' son primos entre sí.

Demostración: Si c es un máximo común divisor de a' y b', entonces

$$a = d.a' = d.(c.a_1) = (d.c).a_1$$
$$b = d.b' = d.(c.b_1) = (d.c).b_1$$

, y así, $d.c$ es un divisor común de a y b. Al ser d un máximo común divisor de estos elementos,

$$d = (d.c).q \quad \Leftrightarrow \quad d.e = d.(c.q).$$

Según la regla de simplificación, de aquí resulta $c.q = e$ y así, c es un elemento invertible (asociado con e).

Teorema 3.8: Si $a,b,t \in G$, $d = (a,b)$ y existe un máximo común divisor de $t.a$ y $t.b$, entonces $d.t$ es un máximo común divisor de $t.a$ y $t.b$.

Demostración: Si d' es un máximo común divisor de $t.a$ y $t.b$, entonces teniendo en cuenta las igualdades siguientes,

$$t.a = t.(d.a') = (t.d).a' \qquad (3.1)$$
$$t.b = t.(d.b') = (t.d).b' \qquad (3.2)$$

, resulta que $t.d$ es un divisor común de $t.a$ y $t.b$ y así, será un divisor también de d'.

Por tanto, $d' = t.d.q$ y dado que d' es un máximo común divisor de $t.a$ y $t.b$,

$$t.a = d'.q_1 = t.d.q.q_1 \qquad (3.3)$$
$$t.b = d'.q_2 = t.d.q.q_2 \qquad (3.4)$$

Combinando (3.3) con (3.1) y (3.4) con (3.2), resulta que

$$(t.d)a' = (t.d).(q.q_1) \qquad (3.5)$$

$$(t.d).b' = (t.d).(q.q_2) \qquad (3.6)$$

, de donde, utilizando la regla de la simplificación, resulta que

$$a' = q.q_1 \quad y \quad b' = q.q_2 .$$

Por tanto, q es un divisor común de a' y b', primos entre sí según el teorema 3.7. Así q es un divisor de un elemento invertible y por tanto q mismo es un elemento invertible.

Así, según el teorema 3.4, $t.d$ será un máximo común divisor de $t.a$ y $t.b$.

Teorema 3.9: Si dos elementos cualesquiera del semigrupo G tienen máximo común divisor, entonces existe también el mínimo común múltiplo de dos elementos cualesquiera y $(a,b).[a,b]$ está asociado con el producto $a.b$.

Demostración: Si $d = (a,b), a = d.a',\ b = d.b'\ y\ t = d.a'.b'$ es obvio que t es un múltiplo común de a y b. Sea ahora t' un múltiplo común cualquiera de a y b. Entonces,

$$t' = a.q_1 = d.a'.q_1$$

$$t' = b.q_2 = d.b'.q_2$$

, de donde resulta que

$$b'.t' = d.a'.b'.q_1 = t.q_1$$

$$a'.t' = d.a'.b'.q_2 = t.q_2$$

, y así t es un divisor común de $a'.t\ y\ b'.t$. Por tanto, t es un divisor del máximo común divisor de $a'.t\ y\ b'.t$. Teniendo en cuenta los teoremas 3.7 y 3.8, un máximo común divisor de $a'.t\ y\ b'.t$ debe tener la forma

$$(a',b').t' = u.t'$$

, donde u es un elemento invertible. Así pues, t es un divisor de $t'.u$ y así, existirá $s \in G$ tal que

$$t'.u = t.s \quad \Leftrightarrow \quad t' = t.s.u^{-1}$$

así, t es un divisor de t' y por tanto, t es un mínimo común múltiplo de a y b.

Observación 3.7: Si en el semigrupo G el elemento neutro es el único elemento invertible y existe el máximo común divisor de dos elementos cualesquiera, entonces $(a,b)\ y\ [a,b]$ son únicas y

$$(a,b).[a,b] = a.b \qquad (3.7)$$

, cualesquiera que sean $a,b \in G$.

Teorema 3.10: Si los elementos a y b están asociados con los elementos a' y b', respectivamente, y d (d') es un máximo común divisor de a y b (de a' y b'), entonces d y d' están asociados.

Demostración: Si a' está asociado con a y b' con b, entonces

$$a' = a.u \quad y \quad b' = b.u_2$$

, donde u_1 y u_2 son dos elementos invertibles. Puesto que d' es un máximo común divisor de a' y b',

$$a' = d'.q_1 \quad y \quad b' = d'.q_2$$

, es decir,

$$a.u_1 = d'.q_1 \quad y \quad b.u_2 = d..q_2$$

Así,

$$a = d'.(q_1.u_1^{-1}) \quad y \quad b = d'.(q_2.u_2^{-1})$$

, y por tanto, d' es un divisor común de a y b. Puesto que d es un máximo común divisor de a y b, d' es un divisor de d.

Por otra parte, d es un máximo común divisor de a y b y así

$$a = d.r_1 \quad y \quad b = d.r_2$$

, de donde resulta que

$$a' = a.u_1 = d.(r_1.u_1) \quad y \quad b' = b.u_2 = d.(r_2.u_2)$$

De esta manera, d es un divisor común de a' y b' y puesto que d' es un máximo común divisor de estos

mismos elementos, d será un divisor de d'. Ahora, según el teorema 3.1, resulta que d y d' están asociados.
La demostración del teorema siguiente es análoga con la del teorema anterior.
Teorema 3.11: Si los elementos a y b están asociados con los elementos a' y b', respectivamente, y m (m') es un mínimo común múltiplo de a y b (de a' y b'), entonces m y m' están asociados.
El máximo común divisor y el mínimo común múltiplo de n elementos del semigrupo G se define de la misma manera que en el caso de dos elementos.
Definición 3.9: Si $a_1, a_2, \dots a_n \in G$, $d \in G$ es un máximo común divisor de $a_1, a_2, \dots a_n$ si d es un divisor común de estos elementos y cualquier otro divisor común de ellos es un divisor de d.
Definición 3.10: Si $a_1, a_2, \dots a_n \in G$, $m \in G$ es un mínimo común múltiplo de $a_1, a_2, \dots a_n$ si m es un múltiplo común de estos elementos y cualquier otro múltiplo común es un múltiplo de m.
Obviamente, como en el caso de dos elementos, el máximo común divisor y el mínimo común múltiplo de n elementos son determinados hasta un factor invertible. Así,

$$(a_1, a_2, \dots a_n) \quad \text{y} \quad [a_1, a_2, \dots a_n]$$

, designarán un máximo común divisor y un mínimo común múltiplo cualquiera de los elementos $a_1, a_2, \dots a_n$ respectivamente.
Teorema 3.12: Si $n \in N - \{0,1\}$ y dos elementos cualesquiera del semigrupo **G** tienen máximo común divisor, entonces

$$(a_1, a_2, \dots a_n) \quad y \quad [a_1, a_2, \dots a_n]$$

, existen cualesquiera que sean $a_1, a_2, \dots a_n \in G$.
Demostración: Primero, si dos elementos cualesquiera de G tienen máximo común divisor, entonces, según el teorema 3.9, tienen también mínimo común múltiplo.
Si $2 \le k \le n-1$, sean d_k y m_k definidos por recurrencia de la manera siguiente:

$$d_2 = (a_1, a_2);\ d_{k+1} = (d_k, a_{k+1}) \tag{3.8}$$

$$m_2 = [a_1, a_2];\ m_{k+1} = [m_k, a_{k+1}] \tag{3.9}$$

De (3.8) resulta que

$$a_1 D d_2,\ d_i D d_n \quad y \quad a_i D d_i$$

, donde $i \in [2, n]$. Por tanto $d_i D d_n$, cualquiera que sea $i \in [1, n]$.
Si d_n' es un divisor común cualquiera de $a_1, a_2, \dots a_n$, entonces se puede demostrar por recurrencia:

$$d_i D d_n' \quad (2 \le i \le n-1) \ y \ a_n D d_n' \quad \Rightarrow \quad d_n D d_n'$$

, y por tanto d_n es un máximo común divisor de $a_1, a_2, \dots a_n$.
De manera análoga resulta que m_n es un mínimo común múltiplo de $a_1, a_2, \dots a_n$. Si sabemos calcular el máximo común divisor y el mínimo común múltiplo de dos elementos, entonces (3.8) y (3.9) permiten calcular el máximo común divisor y el mínimo común múltiplo de n elementos y este proceso es fácilmente programable. Evidentemente, una permutación de los elementos $a_1, a_2, \dots a_n$ entre si cambiaría los resultados solo en un factor invertible.
Definición 3.11: $a \in G$ es un elemento irreducible si no es invertible y no tiene otros divisores que los elementos invertibles o los elementos asociados con a.
Definición 3.12: $p \in G$ es un elemento primo si no es invertible y

$$a.b D p \quad \Rightarrow \quad aDp \ ó \ bDp.$$

Teorema 3.13: Cualquier elemento primo es irreducible.
En efecto, si $p = a.b$, entonces de pDp resulta que $a.bDp$ y así tendremos aDp ó bDp, lo que implica que o bien a o bien b está asociado con p. Si a (b) está asociado con p entonces b (a) es un elemento invertible.
La recíproca de este teorema no es verdadera.
Teorema 3.14: Si a es un elemento irreducible del semigrupo G y a no es un divisor de b, entonces a y b son primos entre sí.
Demostración: Al no ser a un divisor de b, ningún elemento asociado con a puede ser un divisor de b.

Sea ahora d un divisor común de a y b. Al ser d un divisor del elemento irreducible a, d es o bien un elemento invertible o bien es un elemento asociado con a. Pero, teniendo en cuenta que d es también un divisor de b, d no puede estar asociado con a, ya que a no es un divisor de b. Entonces, cualquier divisor común d de a y b es un elemento invertible. Por consiguiente, a y b son primos entre sí.

Teorema 3.15: Si dos elementos cualesquiera del semigrupo G tienen máximo común divisor, entonces cualquier elemento irreducible es un elemento primo.

Demostración: Si c es un elemento irreducible y $a.b\,D\,c$, entonces, según el teorema 3.13, de la falsedad de aDc resultaría que el elemento neutro e es un máximo común divisor de los elementos c y a. Según el teorema 3.8, $b = b.e$ es un máximo común divisor de $c.b$ y $a.b$ y, dado que c es un divisor común de $c.b$ y $a.b$, resulta que c es un divisor de b.

Teorema 3.16: Dos elementos a y b irreducibles y no asociados son primos entre sí.

Demostración: El elemento a no es un divisor de b puesto que al ser b un elemento irreducible, $b = a.q$ sería posible solo para q invertible y en este caso a y b estarían asociados. De esta manera, el teorema 3.16 es consecuencia del teorema 3.14.

Teorema 3.17: Dos elemento primos no asociados a y b son primos entre si.

Demostración: Si a y b son dos elementos primos entre sí pero no están asociados, entonces, según el teorema 3.13, los elementos a y b son irreducibles y no asociados. Así, el teorema 3.17 es consecuencia del teorema 3.16.

Teorema 3.18: Si el elemento a está asociado con un elemento irreducible b entonces a es irreducible.

Demostración: En efecto, $b = a.u$, donde u es un elemento invertible. Suponiendo que b no es irreducible se podría escribir que $b = q.r$, donde q y r no son invertibles y por tanto, tampoco están asociados con b. Así resultaría que

$$a = b.u^{-1} = q.r.u^{-1}$$

El elemento $r.u^{-1}$ no puede ser invertible puesto que en este caso r sería también invertible. De esta manera, q es un divisor no es invertible de a y tampoco está asociado con a, es decir, a no es irreducible. La contradicción obtenida demuestra que b debe ser irreducible.

Teorema 3.19: Un elemento asociado con un elemento primo es primo.

Demostración: Si p es un elemento primo y $p' = p.u$, donde u es un elemento invertible, supongamos que $a.b\,D\,p'$. Entonces,

$$a.b = p.u.q$$

Dado que p es un elemento primo, resulta que aDp ó bDp, es decir aDp' ó bDp'.

§. 4. Divisibilidad en el conjunto de los números naturales.

Sea **N** el conjunto de los números naturales. Con respecto la multiplicación, $N - \{0\}$ es un semigrupo conmutativo con el elemento neutro 1, donde se cumple la regla de simplificación.

Así, en el semigrupo $(N - \{0\}, .)$ quedarán válidas todas las consideraciones del párrafo anterior. El número 1 siendo el único elemento invertible del semigrupo $(N - \{0\}, .)$, dos elementos de este semigrupo no pueden tener más de un máximo común divisor o más de un mínimo común múltiplo.

La estructura algebraica de N es mucho más compleja que la de un semigrupo. En N se definen tres leyes de composición internas: la adición, la multiplicación y una sustracción no siempre posible. $(N, .)$ es un semigrupo multiplicativo con el elemento neutro 1 y $(N, +)$ es un semigrupo aditivo con el elemento neutro 0. La multiplicación es distributiva respecto la adición y la sustracción. Finalmente, existe en N una relación de orden respecto al cual N queda totalmente y bien ordenado. Esto quiere decir que dos elementos cualesquiera de N son comparables y que cualquier subconjunto no vacío de N tiene un elemento más pequeño.

Estas circunstancias permiten demostrar gran cantidad de teoremas sobre la divisibilidad en N que no se podrían formular o demostrar con los medios disponibles en el semigrupo $(N, .)$.

Teorema 4.1: Si $d \in N$ es un divisor común de $a, b \in N$, entonces d es un divisor de $a + b$. En efecto, de

$a = d.a_1$ y $b = d.b_1$, resulta que

$$a + b = d.a_1 + d.b_1 = d.(a_1 + b_1)$$

Teorema 4.2: Si $d \in N$ es un divisor de $a \in N$, entonces d es un divisor de $a.b$, cualquiera que sea $b \in N$. En efecto, puesto que $a = d.a_1$,

$$a.b = (d.a_1).b = d.(a_1.b)$$

Teorema 4.3: Si $a,b,c \in N$, $a = b + c$ y $d \in N$ es un divisor de a y b, entonces d es un divisor de c. En efecto, de $a = d.a_1, b = d.b_1$ y $a = b + c$ resulta que

$$d.a_1 = d.b_1 + c$$

, y puesto que $a \geq b \Rightarrow a_1 \geq b_1$, tendremos $c = d.(a_1 - b_1)$

Teorema 4.4: Si $a,b \in N$ y $b \neq 0$, entonces existen $q,r \in N$ tales que

$$a = b.q + r \;\; y \;\; 0 \leq r < b$$

Demostración: Si $a < b$, entonces

$$a = b.0 + a\,, \;\; donde\, a < b$$

Si $a = b$, entonces

$$a = b.1 + 0\,, \; donde \;\; 0 < b$$

Si $a > b$, sea

$$M = \{n \,/\, n \in N, a < b.n\} \subset N\,.$$

Puesto que $b \neq 0, b = 1 + \alpha$, donde $\alpha \geq 0$. Entonces

$$b.(a+1) = a + \alpha.(a+1) > a$$

, y así $(a+1) \in M$. Por tanto, M no es el conjunto vacío.

Puesto que cualquier subconjunto no vacío de N posee un elemento más pequeño, sea n_0 el elemento más pequeño de M. Dado que $a > b$ y $n_0 \neq 0$, resulta que $n_0 > 1$. Si se pone $q_0 = n_0 - 1$, entonces $q_0 \in M$ y, por consiguiente $a \geq b.q_0$, es decir $a - b.q_0 \geq 0$. La desigualdad $a - b.q_0 \geq b$ es imposible puesto que esto implicaría $a \geq b.n_0$. Así pues

$$a - b.q_0 = r\,, \text{ donde } 0 \leq r < b\,.$$

Observación 4.1: Si $a \in N$, el máximo común divisor de a y 0 es a, puesto que a es un divisor común de a y 0, y cualquier divisor común de a y 0 es un divisor de a.

Observación 5.2: Si $a \in N$, el mínimo común múltiplo de a y 0 es 0 puesto que 0 es un múltiplo común de a y 0, y cualquier múltiplo de a y 0 es cero (por tanto un divisor de 0).

Teorema 4.5: Dos elementos cualesquiera de $(N,.)$ tienen máximo común divisor.

Demostración: Teniendo en cuenta la observación (4.1), es suficiente demostrar el teorema para $a,b \in N - \{0\}$. Según el teorema 5.4, existirán $q_1, r_1 \in N$ tales que

$$a = b.q_1 + r_1 \;\; y \;\; r_1 < b\,.$$

Si $r_1 \neq 0$, existirán $q_2, r_2 \in N$ tales que

$$b = r_1.q_2 + r_2 \;\; y \;\; r_2 < r_1\,.$$

Si $r_2 \neq 0$, existirán $q_3, r_3 \in N$ tales que

$$r_1 = r_2.q_3 + r_3 \;\; y \;\; r_3 < r_2\,.$$

Si $r_3 \neq 0$, se continúa el proceso anterior. No obstante, este proceso no puede continuarse indefinidamente puesto que en el caso contrario se llegaría a una sucesión de números naturales estrictamente decreciente e infinita:

$$b, r_1, r_2, r_3, \ldots$$

, lo que es imposible, al existir solo un número finito de números naturales menores que b. Suponiendo que r_{n+1} es el primer resto nulo, tendremos:

$$a = b.q_1 + r_1$$
$$b = r_1.q_2 + r_2$$

$$\begin{aligned} r_1 &= r_2 . q_3 + r_3 \\ &\ldots\ldots\ldots\ldots \\ r_{n-2} &= r_{n-1} . q_n + r_n \\ r_{n-1} &= r_n . q_{n+1} . \end{aligned} \qquad (4.1)$$

Entonces r_n es divisor de r_{n-1} y, según el teorema 4.1, r_n será también divisor de $r_{n-1}.q$. Al ser r_n divisor común de r_n y $r_{n-1}.q_n$, según el teorema 4.1, r_n será divisor también de r_{n-2}. A continuación, al ser r_n divisor común de r_{n-1} y r_{n-2}, resultará que r_n es divisor de r_{n-3} y, repitiendo éste razonamiento un número finito de veces, se llega a la conclusión de que r_n es divisor común de a y b. Al revés, si $d \in N$ es divisor común de a y b, entonces, según los teoremas 4.2 y 4.3, d será divisor de r_1. Al ser d divisor común de b y r_1 , el mismo razonamiento conduce a la conclusión de que d es un divisor de r_2. Repitiendo este razonamiento de un número finito de veces, resultará que d es divisor de r_n.

De esta manera, r_n es divisor común de a y b, y cualquier divisor común de a y b es un divisor de r_n, por tanto, r_n es el máximo común divisor de a y b.

Al demostrar el teorema 4.5, no solo se ha demostrado la existencia del máximo común divisor en $(N,.)$ sino se ha encontrado también el método práctico para su cálculo. Este método se expresa por las fórmulas 4.1 y se llama el algoritmo de Euclides.

Teorema 4.6: Dos elementos cualesquiera de $(N,.)$ tienen mínimo común múltiplo.

Demostración: Teniendo en cuenta la observación 4.2, es suficiente demostrar el teorema para $a,b \in N$. Según el teorema 4.5, dos elementos cualesquiera de $N - \{0\}$ tienen máximo común divisor y entonces el teorema 2.9 garantiza la existencia del mínimo común múltiplo de dos elementos cualesquiera de $N - \{0\}$.

Observación 4.3: Al ser 1 el único elemento invertible en el semigrupo $(N,.)$, el máximo común divisor y el mínimo común múltiplo de dos elementos de N son únicos, y de 2.9 resulta que

$$(a,b).[a,b] = a.b,$$

, cualesquiera que sean $a,b \in N$.

Cuando los números x e y no son demasiado grandes, dentro de un programa Visual-Basic, el máximo común divisor d podría hallarse efectuando la instrucción $d = mcd(x,y)$, donde la función mcd se ha definido de la manera siguiente:

```
Public Function mcd(ByVal a As Double, ByVal b As Double) As Double
        Dim a1 As Double, b1 As Double, c1 As Double
        Dim q As Double, r As Double
        a1 = a: b1 = b
        If a1 < b1 Then
                c1 = a1: a1 = b1: b1 = c1
        End If
        r = 1
        Do Until r = 0
                 q = a1 / b1: r = a1 - b1 * Int(q)
                If r <> 0 Then
                        a1 = b1: b1 = r
                End If
        Loop
        mcd = b1
End Function
```

Naturalmente, el programa tiene que asegurar que los números x e y son enteros.

En Visual-Basic existe la instrucción $a = b \bmod c$. Esto quiere decir que el resto de la división de a entre c es b. Utilizando esta instrucción, la función mcd se podría construir también de la manera siguiente:

```
Public Function mcd(ByVal a As Double, ByVal b As Double) As Double
        Dim a1 As Double, b1 As Double, c1 As Double, r as Double
```

```
        a1 = a: b1 = b
        If a1 < b1 Then
                c1 = a1: a1 = b1: b1 = c1
        End If
        r = 1
        Do Until r = 0
                r=a1 mod b1
                If r <> 0 Then a1 = b1: b1 = r
        Loop
         mcd = b1
End Function
```

Para calcular el mínimo común múltiplo m de dos números naturales x e y, según la observación 4.3, se podría proceder de la manera siguiente: $d = mcd(x, y)$ y $m = (x / d) * y$. La codificación de este cálculo es la siguiente:

```
Public Function mcm(ByVal a As Double, ByVal b As Double) As Double
        Dim d As Double, m As Double, a1 as double
        d = mcd(a,b):a1=a/d:m=a1*b
        mcm = m
End Function
```

Si se trata calcular el máximo común divisor o el mínimo común múltiplo de números naturales grandes se pueden utilizar las funciones *MaxComDiv* y *MinComMult* , expuestas en la monografía: Operaciones con números enteros largos. Sin embargo, aquí se exponen también las versiones más sencillas siguientes:

```
Public Function mcdNg(ByVal a As String, ByVal b As String) As String
        Dim a1 As String, b1 As String, c1 As String
        Dim q As String, r As String, n As Integer
        Dim rr() As String, x(2) As String, dif As String
        a1 = a: b1 = b: n = 14
        x(1) = a1: x(2) = b1: dif = Restar(x(), n)
        If Left$(dif, 1) = "-" Then
                c1 = a1: a1 = b1: b1 = c1
        End If
        r = "1"
        Do Until r = "0" Or r = "-0"
                x(1) = a1: x(2) = b1: rr() = DivisionEuclidea(x(), n)
                q = rr(1): r = rr(2)
                If r <> "0" Then
                        a1 = b1: b1 = r
                End If
         Loop
    mcdNg = b1
End Function

Public Function mcmNg(ByVal a As String, ByVal b As String) As String
        Dim n As Integer, x(2) As String, d As String
        Dim rr() As String, pr As String
         n = 7: x(1) = a: x(2) = b: d = mcdNg(a, b)
        x(1) = a: x(2) = b: pr = Multiplicar(x(), n)
        x(1) = pr: x(2) = d
         rr() = DivisionEuclidea(x(), 30)
        mcmNg = rr(1)
End Function
```

Puesto que dos números naturales siempre tienen máximo común divisor, según los teoremas 4.13 y 4.15, los números primos coinciden con los números irreducibles. Así, para averiguar que un número natural n es primo, hay que demostrar que este número no tiene otros divisores que 1 y n. Esto se podría hacer dividiendo el número n sucesivamente con los números naturales situados entre 2 y $n-1$, ambos inclusive. Si al efectuar estas divisiones, en cierto momento la división es exacta, el número no es primo. En el caso contrario, el número será primo. El teorema siguiente reducirá considerablemente el número de las divisiones necesarias, para poder decidir si un número es primo o no.

Teorema 4.7: Si el número natural n tiene un divisor mayor que $\sqrt{n}$, entonces tendrá también un divisor menor que $\sqrt{n}$.
En efecto, si p es un divisor de n y p cumple la condición

$$p > \sqrt{n} \qquad (4.2)$$

, entonces teniendo en cuenta que $n = p.q$ y multiplicando ambas partes del desigualdad 4.2 con q, se obtiene que

$$n > q.\sqrt{n}$$

, es decir, para el divisor q de n se cumple la desigualdad:

$$\sqrt{n} > q$$

Observación 4.4: Para averiguar si el número n es primo, según el teorema anterior, es bastante probar que no tiene ningún divisor p cumpliendo las desigualdades

$$2 \le p \le \sqrt{n}$$

Utilizando la observación 4.4, para averiguar si el número contenido en la variable ax (de precisión doble) es primo o no, se puede utilizar la función siguiente.

```
Public Function primoA(ByVal xx As Double) As String
        Dim x1 As Double, x2 As Double, xi As Double
        Dim i As Double, rx As String
        Select Case xx
        Case 1
                rx = "El número no es primo."
        Case 2
                 rx = "El número es primo."
        Case Else
                x2 = xx / 2
                If Int(x2) = x2 Then
                        rx = "El número no es primo." + rc + Str$(2) + " es un divisor."
                Else
                        i = 3: x1 = Sqr(xx)
                        Do While i <= x1
                                xi = xx / i
                                If Int(xi) = xi Then
                                        rx = "El número no es primo." + rc + Str$(i) + " es un divisor."
                                        Exit Do
                                Else
                                        i = i + 2
                                End If
                        Loop
                        If i > x1 Then
                                rx = "El número es primo."
                         End If
                End If
        End Select
        primoA = rx
End Function
```

Si se quiere averiguar si un número natural muy largo es primo o no, se puede utilizar la función $\Pr imoB$, que puede operar con números naturales muy grandes.

```
Public Function primoB(ByVal xx As String) As String
        Dim xa As String, x1 As String, rr() As String
        Dim x(2) As String, xi As String, i As String
        Dim dif As String, n As Integer, rx As String
        n = 7: rc = Chr$(13) + Chr$(10): xa = xx
         If Left$(xa, 1) = " " Then xa = Right$(xa, 1): xa = xx
        If xa = "1" Then
                rx = "El número no es primo.": primoB = rx: Exit Function
        End If
        If xa = "2" Or xa = "3" Or xa = "5" Or xa = "7" Then
                rx = "El número es primo."
                primoB = rx: Exit Function
        End If
        x(1) = xa: x(2) = "2"
        rr() = DivisionEuclidea(x(), n)
        If rr(2) = "0" Or rr(2) = "-0" Then
                rx = "El número no es primo." + rc + Str$(2) + " es un divisor."
                primoB = rx: Exit Function
```

```
        End If
        i = "3": x(1) = xx: x(2) = "2": x1 = RaizIndiceNDecPos(x(), 1, n)
        Do
                x(1) = x1: x(2) = i: dif = RestarDec(x(), n)
                If Left$(dif, 1) <> "-" Then
                        x(1) = xa: x(2) = i: rr() = DivisionEuclidea(x(), n)
                        If rr(2) = "0" Or rr(2) = "-0" Then
                                rx = "El número no es primo." + rc + i + " es un divisor.": Exit Do
                        End If
                        x(1) = i: x(2) = "2": i = Sumar(x(), n)
                        x(1) = i: x(2) = x1: dif = Restar(x(), n)
                        If Left$(dif, 1) <> "-" Then
                                rx = "El número es primo.": Exit Do
                        End If
                End If
        Loop
        primoB = rx
End Function
```

Por ejemplo, utilizando la función $\Pr imosB$ el número $9371256771\,395371$ no es primo , puesto que 263 es un divisor suyo. Para averiguar si el número $3172459973\,1157371$ es primo o no, la función $\Pr imosA$ no sirve ya que el número es demasiado grande. Utilizando la función $\Pr imosB$, resulta que 23 es un divisor de este número y así no es primo. Utilizando cualquiera de las funciones $\Pr imosA$ o $\Pr imosB$, resulta que el número $2537834557\,1$ es primo.

Observación 4.5: Si el número natural n no tiene ningún divisor primo menor o igual que $\sqrt{n}$ entonces tampoco tendrá divisores menores o iguales que $\sqrt{n}$. Por tanto, según la observación 4.4, n será un número primo.

Observación 4.6: Si n es un número natural cualquiera, según el teorema 4.4, existirán $k, r \in N$ tales que

$$n = 6k + r, \text{ donde } 0 \le r < 6.$$

Obviamente, si n es un número primo mayor que 3, r no puede tomar los valores $0, 2, 3\ y\ 4$. Por tanto los números primos mayores que 3 tienen la forma $6k+1$ ó $6k+5$, donde $k \in N$. Dado que

$$6k + 5 = 6(k+1) - 1 = 6k' - 1, \quad (k' > 1)$$

, los números primos mayores que 3 tendrán la forma $6k \pm 1$, donde $k \in N^*$. Obviamente, no todos los números de esta forma son primos. Por ejemplo si $k = 4$ entonces $6k + 1 = 25$ no es primo, pero $6k - 1 = 23$ lo es. Si $k = 7$, ninguno de los números $6k \pm 1$ es primo. Por tanto, un número p diferente de 2 y de 3 es primo si no es divisible con 2, ni con 3, y con ninguno de los números de la forma $6k \pm 1\ (k = 1, 2, \ldots)$, menores que $\sqrt{p}$.

Teniendo en cuenta la observación 4.6 las funciones $\Pr imosA$ y $\Pr imosB$ se pueden modificar de la manera siguiente:

```
Public Function primoC(ByVal xx As Double) As String
        Dim x1 As Double, xxd As Double, rx As String
        Dim i As Double, j As Double, k As Double
        Select Case xx
                Case 1
                        rx = "El número no es primo."
                Case 2, 3
                        rx = "El número es primo."
                Case 4
                        rx = "El número no es primo. 2 es un divisor."
                Case Else
                        xxd = xx / 2
                        If Int(xxd) = xxd Then
                                primoC = "El número no es primo." + rc + "2" + " es un divisor."
                                Exit Function
                        End If
                        xxd = xx / 3
                        If Int(xxd) = xxd Then
                                primoC = "El número no es primo." + rc + "3" + " es un divisor."
                                Exit Function
                        End If
                        x1 = Sqr(xx): k = 1
                        Do
                                i = 6 * k: j = i - 1
```

```
                    If j < x1 Then
                        xxd = xx / j
                        If Int(xxd) = xxd Then
                            rx = "El número no es primo." + rc + Str$(j) + " es un divisor."
                            Exit Do
                        End If
                    End If
                    j = i + 1
                    If j < x1 Then
                        xxd = xx / j
                        If Int(xxd) = xxd Then
                            rx = "El número no es primo." + rc + Str$(j) + " es un divisor."
                            Exit Do
                        End If
                    End If
                    If i < x1 Then
                            k = k + 1
                    Else
                        rx = "El número es primo."
                        Exit Do
                    End If
                Loop
        End Select
        primoC = rx
End Function
'-----------------------------------------------------------------------------------------------------------------------------------
Public Function primoD(ByVal xx As String) As String
        Dim x1 As String, rx As String, rc As String
        Dim i As String, j As String, k As String, x(2) As String
        Dim dif As String, rr() As String, n As Integer
         n = 7: rc = Chr$(13) + Chr$(10)
        Select Case xx
            Case "1"
                rx = "El número no es primo."
            Case "2", "3"
                rx = "El número es primo."
            Case "4"
                rx = "El número no es primo. 2 es un divisor."
            Case Else
                x(1) = xx: x(2) = "2"
                rr() = DivisionEuclidea(x(), n)
                If rr(2) = "0" Or rr(2) = "-0" Then
                    primoD = "El número no es primo." + rc + "2" + " es un divisor."
                    Exit Function
                End If
                x(1) = xx: x(2) = "3"
                rr() = DivisionEuclidea(x(), n)
                If rr(2) = "0" Or rr(2) = "-0" Then
                    primoD = "El número no es primo." + rc + "3" + " es un divisor."
                    Exit Function
                End If
                x(1) = xx: x(2) = "2"
                x1 = RaizIndiceNDecPos(x(), 1, n): k = 1
                Do
                    x(1)= "6": x(2) = k : i = Multiplicar(x(),n)
                    x(1) = i: x(2) = "1": j = Restar(x(), n)
                    x(1) = j: x(2) = x1: dif = Restar(x(), n)
                    If Left$(dif, 1) = "-" Then
                        x(1) = xx: x(2) = j: rr() = DivisionEuclidea(x(), n)
                        If rr(2) = "0" Or rr(2) = "-0" Then
                            rx = "El número no es primo." + rc + j + " es un divisor."
                            Exit Do
                        End If
                    End If
                    x(1) = i: x(2) = "1": j = Sumar(x(), n)
                    x(1) = j: x(2) = x1: dif = Restar(x(), n)
                    If Left$(dif, 1) = "-" Then
                        x(1) = xx: x(2) = j: rr() = DivisionEuclidea(x(), n)
                        If rr(2) = "0" Or rr(2) = "-0" Then
                            rx = "El número no es primo." + rc + j + " es un divisor."
                            Exit Do
                        End If
                    End If
                    x(1) = i: x(2) = x1: dif = Restar(x(), n)
                    If Left$(dif, 1) = "-" Then
                        x(1) = k: x(2) = "1": k = Sumar(x(), n)
                    Else
                        rx = "El número es primo."
                        Exit Do
                    End If
                Loop
```

```
        End Select
        primoD = rx
End Function
```

Como ejercicio es útil que, teniendo a la vista las funciones $\Pr imosA$ y $\Pr imosC$ el lector transforme estas funciones en las funciones $\Pr imosB$ y $\Pr imosD$, respectivamente y luego haga la comparación de lo obtenido con las funciones expuestas.

Para calcular los primeros n_0 números primos se proponen las funciones siguientes:

```
Public Function PrimerosNPrimosA(n0 As Double) As Variant
        Dim i As Double, k As Double, m As Double, sw As Integer, j As Double
        Dim x As Double, x1 As Double, q As Double, p0() As Double
        ' Cálculo de los primeros n0 números primos
         ' Los números primos se colocan en la matriz P0()
        ReDim p0(n0)
        p0(1) = 2: p0(2) = 3
        i = 1: k = 1: j = 3: m = 0: x = 6 * k - 1 + m
        Do Until j = n0 + 1
                x1 = Int(Sqr(x))
                Do Until p0(i) > x1 Or sw = 1
                        q = x / p0(i)
                        If Int(q) <> q Then i = i + 1 Else sw = 1
                Loop
                If sw = 0 Then p0(j) = x: j = j + 1 Else sw = 0
                If m = 2 Then m = 0: k = k + 1 Else m = 2
                x = 6 * k - 1 + m: i = 1
        Loop
        PrimerosNPrimosA = p0()
End Function
'-----------------------------------------------------------------------------------------------------------
Public Function PrimerosNPrimosB(ByVal n0 As String) As Variant
        Dim i As String, k As String, m As String, sw As Integer, n As Integer
        Dim xx As String, x1 As String, q As String, p0() As String, rr() As String
        Dim x(2) As String, n1 As String, dif As String, dx1i As String, j As String
         ' Cálculo de los primeros n0 números primos
         ' Los números primos se colocan en la matriz P0()
        ReDim p0(n0)
        n = 7: p0("1") = "2": p0("2") = "3"
        i = "1": k = "1": m = "0": j = "3": x(1) = "6": x(2) = k: x(1) = Multiplicar(x(), n)
        x(2) = "1": x(1) = Restar(x(), n): x(2) = m: xx = Sumar(x(), n)
        x(1) = n0: x(2) = "1": n1 = Sumar(x(), n)
        Do Until j = n1
                x(1) = xx: x(2) = "2": x1 = RaizIndiceNDecPos(x(), 2, n)
                Do
                        x(1) = x1: x(2) = p0(i): dx1i = RestarDec(x(), n)
                        If Left$(dx1i, 1) = "-" Or sw = 1 Then Exit Do
                        x(1) = xx: x(2) = p0(i): rr() = DivisionEuclidea(x(), n)
                        If rr(2) <> "0" Then
                                x(1) = i: x(2) = "1": i = Sumar(x(), n)
                        Else
                                sw = 1
                        End If
                Loop
                If sw = 0 Then p0(j) = xx: x(1) = j: x(2) = "1": j = Sumar(x(), n) Else sw = 0
                If m = "2" Then m = "0": x(1) = k: x(2) = "1": k = Sumar(x(), n) Else m = "2"
                x(1) = "6": x(2) = k: x(1) = Multiplicar(x(), n)
                x(2) = "1": x(1) = Restar(x(), n): x(2) = m: xx = Sumar(x(), n): i = "1"
        Loop
        PrimerosNPrimosB = p0()
End Function
```

Si se pone $n_0 = 639$, se calcularán los primeros 639 números primos:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97,101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283 , 293, 307,311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 953, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997, 1009, 1013, 1019, 1021, 1031, 1033, 1039,

1049, 1051, 1061, 1063, 1069, 1087, 1091,1093, 1097, 1103, 1109, 1117, 1123, 1129, 1151, 1153, 1163, 1171, 1181, 1187, 1193, 1201, 1213, 1217, 1223, 1229, 1231, 1237, 1249, 1259, 1277, 1279, 1283, 1289, 1291,1297, 1303, 1307, 1319, 1321, 1327, 1361, 1361, 1367, 1373, 1381, 1399, 1409, 1423, 1427, 1429, 1433, 1439, 1447, 1451, 1453, 1459, 1471, 1481, 1483, 1487, 1489, 1493, 1499, 1511, 1523, 1531, 1543, 1549, 1553, 1559, 1567, 1571, 1579, 1583, 1597, 1601, 1607, 1609, 1613, 1619, 1621, 1627, 1637, 1657, 1663, 1667, 1669, 1693, 1697, 1699, 1709, 1721, 1723, 1733, 1741, 1747, 1753, 1759, 1777, 1783, 1787, 1789, 1801, 1811, 1823, 1831, 1847, 1861, 1867, 1871, 1873, 1877, 1879, 1889, 1901, 1907, 1913, 1931, 1933, 1949, 1951, 1973, 1979, 1987, 1993, 1997, 1999, 2003, 2011, 2017, 2027, 2029,2039, 2053, 2063, 2069, 2081, 2083, 2087, 2089, 2099, 2111, 2113, 2129, 2131, 2137, 2141, 2143, 2153, 2161, 2179, 2203, 2207, 2213, 2221, 2237, 2239, 2243, 2251, 2267, 2269, 2273, 2281, 2287, 2293, 2297, 2309, 2311, 2333, 2339, 2341, 2347, 2351, 2357, 2371, 2377, 2381, 2383, 2389, 2393, 2399, 2411, 2417, 2423, 2437, 2441, 2447, 2459,2467, 2473, 2477, 2503, 2521, 2531, 2539, 2543, 2549, 2551, 2557, 2579, 2591, 2593, 2609, 2617, 2621, 2633, 2647, 2657, 2659, 2663, 2671, 2677, 2683, 2687, 2689, 2693, 2699, 2707, 2711, 2713, 2719, 2729, 2731, 2741, 2749, 2753, 2767, 2777, 2789, 2791,2797, 2801, 2803, 2819, 2833, 2837, 2843, 2851, 2857, 2861, 2879, 2887, 2897, 2903,2909, 2917, 2927, 2939, 2953, 2957, 2963, 2969, 2971, 2999, 3001, 3011, 3019, 3023, 3037, 3041, 3049, 3061, 3067, 3079, 3083, 3089, 3109, 3119, 3121, 3137, 3163, 3167, 3169, 3181, 3187, 3191, 3203, 3209, 3217, 3221, 3229, 3251, 3253, 3257, 3259, 3271,3299, 3301, 3307, 3313, 3319, 3323, 3329, 3331, 3343, 3347, 3359, 3361, 3371, 3373, 3389, 3391, 3407, 3413, 3433, 3449, 3457, 3461, 3463, 3467, 3469, 3491, 3499, 3511, 3517, 3527, 3529, 3533, 3539, 3541, 3547, 3557, 3559, 3571, 3581, 3583, 3593, 3607, 3613, 3617, 3623, 3631, 3637, 3643, 3659, 3671, 3673, 3677, 3691, 3697, 3701, 3709,3719, 3727, 3739, 3761, 3767, 3769, 3779, 3793, 3797, 3803, 3821, 3823, 3833, 3847, 3851, 3853, 3863, 3877, 3881, 3889, 3907, 3911, 3917, 3919, 3923, 3929, 3931, 3943, 3947, 3967, 3989, 4001, 4003, 4007, 4013, 4019, 4021, 4027, 4049, 4051, 4057, 4073,4079, 4091, 4093, 4099, 4111, 4127, 4129, 4133, 4139, 4153, 4157, 4159, 4177, 4201, 4211, 4217, 4219, 4229, 4231, 4241, 4243, 4253, 4259, 4261, 4271, 4273, 4283, 4289, 4297, 4327, 4337, 4339, 4349, 4357, 4363, 4391, 4373, 4391, 4397, 4409, 4421, 4423, 4441, 4447, 4451, 4457, 4463, 4481, 4483, 4493, 4507, 4513, 4517, 4519, 4523, 4547,4549, 4561, 4567, 4583, 4591, 4597, 4603, 4621, 4637, 4639, 4643, 4649, 4651, 4657, 4663, 4673, 4679, 4691, 4703, 4721, 4723, 4729

Para hallar los números primos situados en un intervalo dado, se puede utilizar las funciones siguientes:

```
Public Function PrimosEnUnIntervaloA(a1 As Double, a2 As Double) As String
        Dim primos As String, xx As String, i As Double, k As Double, p As Double
        Dim x1 As Double, x2 As Double, q As Double, sw As Integer
        If primos <> "" Then primos = ""
        If a1 <= 2 Then
                primos = primos + "2, 3,"
         Else
                If a1 <= 3 Then
                        primos = primos + "3,"
                End If
        End If
        If a2 >= 5 Then
                k = Int(a1 / 6) + 1: p = 0: x1 = 6 * k - 1 + p
                While x1 <= a2
                        x2 = Int(Sqr(x1)): i = 2: sw = 0
                        Do Until i > x2
                                q = x1 / i
                                If Int(q) <> q Then
                                        i = i + 1
                                Else
                                        sw = 1: Exit Do
                                End If
                        Loop
                        If sw = 0 Then xx = Mid$(Str$(x1), 2): primos = primos + xx + "," Else sw = 0
                        If p = 2 Then k = k + 1: p = 0 Else p = p + 2
                        x1 = 6 * k - 1 + p
                Wend
                If Len(primos) > 1 Then primos = Left$(primos, Len(primos) - 1)
        End If
        If primos = "" Then primos = "¡No hay primos en el intervalo!"
        PrimosEnUnIntervaloA = primos
```

```
End Function
'----------------------------------------------------------------------------------------------------------
Public Function PrimosEnUnIntervaloB(a1 As String, a2 As String) As String
	Dim primos As String, i As String, k As String, p As String
	Dim x1 As String, x2 As String, q As String, sw As Integer
	Dim dif As String, x(2) As String, rr() As String, n As Integer, di As String
	n = 7
	If primos <> "" Then primos = ""
	x(1) = a1: x(2) = "2": dif = Restar(x(), n)
	If dif = "0" Or Left$(dif, 1) = "-" Then
		primos = primos + "2, 3,"
	Else
		x(1) = a1: x(2) = "3": dif = Restar(x(), n)
		If dif = "0" Or Left$(dif, 1) = "-" Then
			primos = primos + "3,"
		End If
	End If
	x(1) = "5": x(2) = a2: dif = Restar(x(), n)
	If dif = "0" Or Left$(dif, 1) = "-" Then
		x(1) = a1: x(2) = "6": rr() = DivisionEuclidea(x(), n)
		x(1) = rr(1): x(2) = "1": k = Sumar(x(), n)
		p = "0": x(1) = k: x(2) = "6": x(1) = Multiplicar(x(), n)
		x(2) = "1": x(1) = Restar(x(), n): x(2) = p: x1 = Sumar(x(), n)
		x(1) = x1: x(2) = a2: dif = Restar(x(), n)
		While dif = "0" Or Left$(dif, 1) = "-"
			x(1) = x1: x(2) = "2": x2 = RaizIndiceNDecPos(x(), 2, n)
			i = "2": sw = 0
			Do
				x(1) = x2: x(2) = i: di = RestarDec(x(), n)
				If Left$(di, 1) = "-" Then Exit Do
				x(1) = x1: x(2) = i: rr() = DivisionEuclidea(x(), n)
				If rr(2) <> "0" Then
					x(1) = i: x(2) = "1": i = Sumar(x(), n)
				Else
					sw = 1: Exit Do
				End If
			Loop
			If sw = 0 Then primos = primos + x1 + "," Else sw = 0
			If p = 2 Then
				x(1) = k: x(2) = "1": k = Sumar(x(), n): p = "0"
			Else
				x(1) = p: x(2) = "2": p = Sumar(x(), n)
			End If
			x(1) = k: x(2) = "6": x(1) = Multiplicar(x(), n): x(2) = "1"
			x(1) = Restar(x(), n): x(2) = p: x1 = Sumar(x(), n)
			x(1) = x1: x(2) = a2: dif = Restar(x(), n)
		Wend
		If Len(primos) > 1 Then primos = Left$(primos, Len(primos) - 1)
		If primos = "" Then primos = "¡No hay primos en el intervalo!"
	End If
	PrimosEnUnIntervaloB = primos
End Function
```

Los números de Mersenne son los términos de la sucesión de término general:

$$M(n) = 2^n - 1 \quad (n \in N^*)$$

Si n es un número compuesto entonces $n = u \cdot v \ (u, v > 1)$. Entonces poniendo $a = 2^u$ en la identidad $a^v - 1 = (a-1)(a^{v-1} + a^{v-2} + \cdots + 1)$, se obtiene que

$$M(n) = M(u \cdot v) = 2^{uv} - 1 = (2^u)^v - 1 = (2^u - 1)(2^{u(v-1)} + 2^{u(v-2)} + 2^u + 1') = u' \cdot v'$$

, donde $u' = 2^u - 1 > 1$ y $v' = 2^{u(v-1)} + \cdots + 2^u + 1 > 1$. Por consiguiente, si n es un número compuesto $M(n)$ lo es también. Así $M(n)$ solo entonces es primo si n es primo, pero pueden existir valores primos de n para los cuales $M(n)$ no es primo. Por ejemplo, calculando $M(29)$ con la función siguiente:

```
Public Function NumMers(ByVal xx As String) As String
	Dim p As String, q As String, x(2) As String, n As Integer
	Dim rc As String, rx As String
	' M(n) = 2^n-1
```

```
        n = 7: rc = Chr$(13) + Chr$(10)
        x(1) = "2": x(2) = xx: p = Potencias(x(), n)
        x(1) = p: x(2) = "1": p = Restar(x(), n)
        q = primoB(p)      ‘  ó bien q = primoD(p)
        rx = "M(" + xx + ") = " + p + rc + q
        NumMers = rx
End Function
```

, resulta que 233 es un divisor de $M(29)$, es decir no es un número primo. Utilizando la función NumMers se obtiene sin dificultad que

$$M(2)=3\ ,\ M(3)=7\ ,\ M(5)=31\ ,\ M(7)=127\ ,\ M(13)=8191$$
$$M(17)=131071\ ,\ M(19)=524287\ \text{ y }\ M(31)=2147483647$$

, son números Mersenne primos. Se verifica también sin dificultad que hasta $M(61)$ no hay otros números Mersenne primos. El número Mersenne $M(61)$ es primo pero la verificación ya necesita su tiempo. A partir de este momento la verificación de que un número Mersenne es primo ya dura cada vez más, sin embargo es más fácil descartar a aquellos que no son primos.

Los números de Fermat son los términos de la sucesión de término general:

$$F(n)=2^{2^n}+1 \quad (n\in N)$$

Los números de Fermat se pueden calcular por ordenador, utilizando el código siguiente:

```
Public Function NumFermat(ByVal xx As String) As String
        Dim p As String, q As String, x(2) As String
        Dim rc As String, rx As String, n As Integer
         ' F(n)=2^(2^n)+1
         n = 7: rc = Chr$(13) + Chr$(10)
        x(1) = "2":: x(2) = xx: p = Potencias(x(), n)
        x(1) = "2":: x(2) = p: p = Potencias(x(), n)
        x(1) = p: x(2) = "1": p = Sumar(x(), n)
        q = primoB(p)
         rx = "F(" + xx + ")= " + p + rc + q
        NumFermat = rx
End Function
```

Utilizando la función anterior se obtiene que

$$F(0)=3\ ,\ F(1)=5\ ,\ F(2)=17\ ,\ F(3)=257\ ,\ F(4)=65537$$

Al ver que los primeros cinco términos de la sucesión son primos, Fermat pensó esto es así siempre . Sin embargo, según la función NumFermat $F(5)=4294967297$ y tiene el divisor 641 (resultado obtenido por Euler en 1732). Por tanto los números de Fermat no son siempre primos. Luego, a continuación, 274177 es un divisor de $F(6)=18446744073709551617$. A partir de aquí, hallar un divisor para los números

$$F(7)=340282366920938463463374607431768211457$$
$$F(8)=115792089237316195423570985008687907853269984665640394575840079131296399 37$$

F(9) = 13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427600318581864860508537538828119465699464336490060840 97

, es un trabajo que necesita ya mucho tiempo.

Según *Wikipedia*, 59649589127497217 y 1238926361552897 son divisores de $F(7)$ y $F(8)$, respectivamente. Dividir $F(7)$ y $F(8)$ con estos números es fácil si se utiliza la función *DivisionEuclidea* (ver [3]) y se obtienen los cocientes respectivos. Lo que es más difícil, y necesita mucho tiempo, es encontrar un divisor. El lector, si tiene paciencia y tiempo, podría intentar averiguar si $F(9)$ es primo o no. En este momento no se sabe si los primeros cinco números Fermat son los únicos primos o no. Tampoco se sabe si el conjunto de los números primos de Fermat es finito o no.

Una propiedad importante de los números Fermat es la siguiente:

$$\forall\ n\in N,\quad F(n+1)=F(n)\cdot F(n-1)\cdot\ \cdots\ F(0)+2 \qquad (*)$$

En efecto, la igualdad se cumple para $n=0$:

$$F(1)=5=3+2=F(0)+2$$

Suponiendo que la igualdad se cumple para $n = k > 0$ resulta que

$$F(k+2)) = 2^{2^{k+2}} + 1 = 2^{2^{k+1}\cdot 2} - 1 + 2 = \left(2^{2^{k+1}}\right)^2 - 1 + 2 = \left(2^{2^{k+1}} + 1\right)\cdot\left(2^{2^{k+1}} - 1\right) + 2 =$$

$$= F(k+1)\left(2^{2^{k+1}} + 1 - 2\right) + 2 = F(k+1)\cdot\left[F(k+1) - 2\right] = F(k+1)\cdot F(k)\cdot F(k-1)\cdot\cdots\cdot F(0),$$

, es decir la igualdad se cumple para $n = k+1$ también. Por tanto la propiedad (*) se cumple para cualquier número natural.

Luego si $p, q \in N$ y $p > q$ entonces teniendo en cuenta que

$$F(p) = F(p-1)\cdot\cdots\cdot F(q)\cdot\cdots\cdot F(0) + 2$$

, cualquier divisor común r de $F(p)\ y\ F(q)$ debería dividir a 2. Por tanto $r = 1$ ó $r = 2$. Luego, $F(p)\ y\ F(q)$ son números impares y así no pueden tener el divisor 2. Por consiguiente $r = 1$, es decir $F(p)\ y\ F(q)$ son primos entre sí.

Dos números primos son gemelos si la diferencia entre el mayor y el menor es 1 ó 2. La diferencia es 1 solo en el caso de de los primos gemelos 2 y 3.

Para hallar los números primos gemelos entre los n_0 primeros números primos, se puede utilizar la función siguiente:

```
Public Function PrimosGemelos(ByVal n0 As String) As Variant
        Dim i As String, k As String, m As String, sw As Integer, n As Integer, j As String
        Dim xx As String, x1 As String, q As String, p0() As String, p() As String, rr() As String
        Dim x(2) As String, n1 As String, n2 As String, dif As String, dx1i As String, rc As String
        Dim swg As Integer, npg As String, j1 As String, ng As String
        ' Cálculo de los números primos gemelos entre los primeros n0 números primos.
        ' Los números primos calculados se colocan en la matriz P0()
         rc = Chr$(13) + Chr$(10): n = 7
        x(1) = n0: x(2) = "1": n2 = Sumar(x(), n)
        ReDim p0(n2), p(n2)
        p0("1") = "2": p0("2") = "3"
        i = "1": k = "1": m = "0": j = "3": x(1) = "6": x(2) = k: x(1) = Multiplicar(x(), n)
        x(2) = "1": x(1) = Restar(x(), n): x(2) = m: xx = Sumar(x(), n)
        'x = 6 * k - 1 + m
        x(1) = n0: x(2) = "1": n1 = Sumar(x(), n)
        Do Until j = n1
                x(1) = xx: x(2) = "2": x1 = RaizIndiceNDecPos(x(), 2, n)
                Do
                        x(1) = x1: x(2) = i: dx1i = RestarDec(x(), n)
                        If Left$(dx1i, 1) = "-" Or sw = 1 Then Exit Do
                                x(1) = xx: x(2) = p0(i): rr() = DivisionEuclidea(x(), n)
                        If rr(2) <> "0" Then x(1) = i: x(2) = "1": i = Sumar(x(), n) Else sw = 1
                Loop
                If sw = 0 Then p0(j) = xx: x(1) = j: x(2) = "1": j = Sumar(x(), n) Else: sw = 0
                If m = "2" Then m = "0": x(1) = k: x(2) = "1": k = Sumar(x(), n) Else m = "2"
                x(1) = "6": x(2) = k: x(1) = Multiplicar(x(), n)
                x(2) = "1": x(1) = Restar(x(), n): x(2) = m: xx = Sumar(x(), n): i = "1"
         Loop
        If swg = 0 Then
                npg = "2,3 ; ": swg = 1: i = "2"
                Do
                        x(1) = i: x(2) = "1": j = Sumar(x(), n)
                        x(1) = p0(j): x(2) = p0(i): dif = Restar(x(), n)
                        If dif = "2" Then npg = npg + p0(i) + "," + p0(j) + " ; "
                        x(1) = i: x(2) = "1": i = Sumar(x(), n)
                        x(1) = i: x(2) = "1": j1 = Sumar(x(), n)
                        If p0(j1) = "" Then Exit Do
                Loop
                If npg <> "" Then npg = Left$(npg, Len(npg) - 2)
        End If
        If npg = "" Then
                ng = "¡Entre los números primos calculados no hay gemelos!"
        Else
                ng = "Los números primos gemelos calculados son:  " + rc
                 ng = ng + npg
        End If
        PrimosGemelos = ng
End Function
```

Para hallar los números primos gemelos en un intervalo $[a,b]$ (donde $a,b>0$) se puede servir de la función siguiente:

```
Public Function PrimosGemelosIntervalo(ByVal a1 As String, ByVal a2 As String) As String
        Dim primos As String, i As String, k As String, p As String, j As Long, xx As String
        Dim x1 As String, x2 As String, q As String, primosg As String, sw As Integer
        Dim dif As String, x(2) As String, rr() As String, n As Integer, di As String
         Dim p1 As String, p2 As String, rt As String, primo1 As String, primo2 As String
         If primos <> "" Then primos = "": n = 7
        x(1) = a1: x(2) = "2": dif = Restar(x(), n)
        If dif = "0" Or Left$(dif, 1) = "-" Then
                 primos = primos + "2,3,"
        Else
                x(1) = a1: x(2) = "3": dif = Restar(x(), n)
                If dif = "0" Or Left$(dif, 1) = "-" Then primos = primos + "3,"

        End If
        x(1) = "5": x(2) = a2: dif = Restar(x(), n)
        If dif = "0" Or Left$(dif, 1) = "-" Then
                x(1) = a1: x(2) = "6": rr() = DivisionEuclidea(x(), n)
                x(1) = rr(1): x(2) = "1": k = Sumar(x(), n)
                p = "0": x(1) = k: x(2) = "6": x(1) = Multiplicar(x(), n)
                x(2) = "1": x(1) = Restar(x(), n): x(2) = p: x1 = Sumar(x(), n)
                x(1) = x1: x(2) = a2: dif = Restar(x(), n)
                While dif = "0" Or Left$(dif, 1) = "-"
                        x(1) = x1: x(2) = "2": x2 = RaizIndiceNDecPos(x(), 2, n)
                        i = "2": sw = 0
                        Do
                                x(1) = x2: x(2) = i: di = RestarDec(x(), n)
                                If Left$(di, 1) = "-" Then Exit Do
                                x(1) = x1: x(2) = i: rr() = DivisionEuclidea(x(), n)
                                If rr(2) <> "0" Then
                                        x(1) = i: x(2) = "1": i = Sumar(x(), n)
                                 Else
                                         sw = 1: Exit Do
                                End If
                        Loop
                        If sw = 0 Then primos = primos + x1 + "," Else sw = 0
                        If p = 2 Then
                                x(1) = k: x(2) = "1": k = Sumar(x(), n): p = "0"
                        Else
                                 x(1) = p: x(2) = "2": p = Sumar(x(), n)
                        End If
                        x(1) = k: x(2) = "6": x(1) = Multiplicar(x(), n): x(2) = "1"
                        x(1) = Restar(x(), n): x(2) = p: x1 = Sumar(x(), n)
                         x(1) = x1: x(2) = a2: dif = Restar(x(), n)
                Wend
                If Len(primos) > 1 Then primos = Left$(primos, Len(primos) - 1)
        End If
        j = 1: primos = primos + ","
        Do
                rt = Right$(Left$(primos, j), 1)
                If rt = "," Then
                        xx = Left$(primos, j - 1): primos = Mid$(primos, j + 1): j = 1: Exit Do
                End If
                j = j + 1
        Loop
        Do
                rt = Right$(Left$(primos, j), 1)
                If rt = "," Then
                        primo1 = xx: primo2 = Left$(primos, j - 1): xx = primo2
                        primos = Mid$(primos, j + 1): j = 1
                        x(1) = primo2: x(2) = primo1: dif = Restar(x(), n)
                        If dif = "2" Or dif = "1" Then
                                primosg = primosg + primo1 + "," + primo2 + "  ;  "
                                If primos = "" Then Exit Do
                        End If
                Else
                         j = j + 1
```

```
            End If
            If primos = "" Then Exit Do
        Loop
        If Right$(primosg, 1) = " " Then primosg = Left$(primosg, Len(primosg) - 3)
        If primosg = "" Then primosg = "¡No hay primos gemelos en el intervalo!"
        PrimosGemelosIntervalo = primosg
End Function
```

§.5. Divisibilidad en un anillo íntegro.

Si $(A,+,\cdot)$ es un anillo íntegro, $(A,\cdot)$ y $(A-\{0\},\cdot)$ son semigrupos conmutativos y con elemento neutro, la regla de simplificación es válida en $(A-\{0\},\cdot)$ pero no en $(A,\cdot)$, puesto que en éste último no se puede simplificar con el elemento 0. Sin embargo en $(A,\cdot)$ se puede simplificar con un elemento no nulo (teorema 1.5). Así pues,

$$a \neq 0 \;\; y \;\; a \cdot b = a \cdot c \;\; \Rightarrow \;\; b = c.$$

Definición 5.1: a es un divisor de b en el anillo $(A,+,\cdot)$ si a es un divisor de b en el semigrupo $(A,\cdot)$.

Definición 5.2: a es un múltiplo de b en el anillo $(A,+,\cdot)$ si a es un múltiplo de b en el semigrupo $(A,\cdot)$.

En el semigrupo $(A,\cdot)$, 0 es un múltiplo de cualquier elemento puesto que $0 = a \cdot 0$, cualquiera que sea $a \in A$. Con otras palabras, 0 es divisible con cualquier elemento del semigrupo $(A,\cdot)$.

Teorema 5.1: Si $d \in A$ es un divisor común de $a,b \in A$, entonces d es un divisor de $a+b$.

Teorema 5.2: Si $d \in A$ es un divisor de $a \in A$, entonces d es un divisor de $a \cdot b$, cualquiera que sea $b \in A$.

Teorema 5.3: Si $a,b,c \in A$, $a = b + c$ y $d \in A$ es un divisor común de a y b, entonces d es un divisor de c.

Las demostraciones de estos últimos tres teoremas son idénticas con las demostraciones correspondientes en el caso de los números naturales (teoremas: 4.1, 4.2, 4.3).

Definición 5.2: Si $(A,+,\cdot)$ es un anillo íntegro y $a,b \in A$, entonces $d \in A$ es un máximo común divisor de $a \; y \; b$ si d es un máximo común divisor de $a \; y \; b$ en el semigrupo $(A,\cdot)$.

Definición 5.3: Si $(A,+,\cdot)$ es un anillo íntegro y $a,b,\in A$, entonces $m \in A$ es un mínimo común múltiplo de a y b si m es un mínimo común múltiplo de a y b en el semigrupo $(A,\cdot)$.

Teorema 5.4: Si A es un anillo íntegro y $a,b \in A-\{0\}$, entonces a es un divisor de b en el semigrupo $(A,\cdot)$ si, y solamente si, a es un divisor de b en el semigrupo $(A-\{0\},\cdot)$.

Demostración: Si a es un divisor de b en $(A,\cdot)$, entonces existe $q \in A$ tal que $b = a \cdot q$. De $b \neq 0$ resulta que $q \neq 0$ y así a es un divisor de b en $(A-\{0\},\cdot)$. El teorema recíproco es obvio. De manera análoga se justifica el teorema siguiente.

Teorema 5.4': Si A es un anillo íntegro y $a,b \in A-\{0\}$ entonces b es un múltiplo de a en el semigrupo $(A,\cdot)$ si, y solamente si, b es un múltiplo de a en el semigrupo $(A-\{0\},\cdot)$.

Teorema 5.5: Si A es un anillo íntegro y $a \in A-\{0\}$ entonces a es un máximo común divisor de a y 0.

Demostración: De las igualdades $a = a \cdot e$ y $0 = a \cdot 0$ resulta que a es un divisor común de a y 0. Por otra parte, si $t \in A$ es un divisor común de a y 0 entonces a es divisible con t y así, a es un máximo común divisor de a y 0.

Teorema 5.6: Si A es un anillo íntegro y $a \in A$, entonces 0 es el único mínimo común múltiplo de a y 0.

Demostración: De las igualdades 0 = a.0 y 0 = 0.0 resulta que 0 es un múltiplo común de a y 0. Cualquier múltiplo común de a y 0 es necesariamente 0 y así es un múltiplo de 0. Por tanto, 0 es un mínimo común múltiplo de a y 0. Evidentemente, no hay otro múltiplo común, puesto que cualquier elemento asociado con 0 es 0.

Teorema 5.7: Si 0 es un máximo común divisor de $a,b \in A$, entonces $a = 0$ y $b = 0$.

En efecto, si 0 es un máximo común divisor de a y b, entonces a y b son múltiplos de 0, y por tanto $a = b = 0$.

Teorema 5.8: Si 0 es un mínimo común múltiplo de $a,b \in A$, entonces $a = 0$ ó $b = 0$.

En efecto, $a \cdot b$ es un múltiplo común de a y b. Al ser 0 un mínimo común múltiplo de a y b , $a \cdot b$ debe

ser un múltiplo de 0 y así $a \cdot b = 0$. Teniendo ahora en cuenta que A es un anillo íntegro, resulta que $a = 0$ ó $b = 0$.

Teorema 5.9: En un anillo íntegro, el máximo común divisor o el mínimo común múltiplo de dos elementos no nulos es no nulo (siempre y cuando existen).

En efecto, en el caso contrario, los teoremas 5.7 y 5.8 implicarían que los dos elementos son nulos o bien que uno de ellos es nulo, según que se trate del máximo común divisor o del mínimo común múltiplo, respectivamente.

Teorema 5.10: Si $a, b \in A - \{0\}$, entonces $d\ (m)$ es un máximo común divisor (un mínimo común múltiplo) de a y b en $(A, \cdot)$ si y solamente si $d\ (m)$ es un máximo común divisor (un mínimo común múltiplo) de a y b en $(A - \{0\}, \cdot)$

El teorema es consecuencia de los teoremas 5.4 y 5.4'.

Observación 5.1: Teniendo en cuenta lo expuesto anteriormente resulta que si se quiere demostrar la existencia del máximo común divisor y del mínimo común múltiplo de dos elementos cualesquiera de un anillo íntegro, entonces la demostración se puede hacer únicamente para los elementos no nulos, puesto que si uno de ellos fuera nulo, ya se sabe que la existencia está asegurada.

Teniendo en cuenta que en el semigrupo $(A - \{0\}, \cdot)$ es válida la regla de simplificación, se pueden aplicar todos los resultados del §.2 para los elementos no nulos de un anillo íntegro A.

Definición 5.4: El anillo A es euclideo si A es un anillo íntegro y existe una aplicación $f : A - \{0\} \to N$, tal que para cualquier $a, b \in A - \{0\}$ se pueda encontrar $q, r \in A$ tales que

$$a = b \cdot q + r \quad (r = 0 \ \text{ó} \ f(r) < f(b)).$$

Teorema 5.11: Dos elementos cualesquiera de un anillo euclideo tienen máximo común divisor y mínimo común múltiplo.

Demostración: Según la observación 5.1, se puede suponer que $a \neq 0$ y $b \neq 0$. Entonces existirán $q_1, r_1 \in A$ tales que

$$a = b \cdot q_1 + r_1 \qquad (r_1 = 0 \ \text{ó} \ f(r_1) < f(b)).$$

Si $r_1 \neq 0$, existirán $q_2, r_2 \in A$ tales que

$$b = r_1 \cdot q_2 + r_2 \ \text{ y } (r_2 = 0 \ \text{ó} \ f(r_2) < f(r_1)).$$

Si $r_2 \neq 0$, se continúa el proceso anterior, pero esto no puede continuarse indefinidamente puesto que la sucesión de números naturales

$$f(b),\ f(r_1),\ f(r_2), \ldots$$

es estrictamente decreciente y solo hay un número finito de números naturales menores que $f(b)$. Así pues, se puede suponer que r_{n+1} es el primer resto nulo en la sucesión de divisiones anteriores y, por tanto:

$$\begin{aligned} a &= b \cdot q_1 + r_1 \\ b &= r_1 \cdot q_2 + r_2 \\ r_1 &= r_2 \cdot q_3 + r_3 \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ r_{n-2} &= r_{n-1} \cdot q_n + r_n \\ r_{n-1} &= r_n \cdot q_{n+1} \end{aligned} \tag{5.1}$$

Siguiendo el mismo razonamiento que en el §.4, en el caso de los números naturales, se llega a la conclusión que r_n es el máximo común divisor de a y b.

Así, dos elementos cualesquiera del semigrupo $(A - \{0\}, \cdot)$ tienen máximo común divisor y entonces, según el teorema 3.9, tendrán también mínimo común múltiplo.

Teorema 5.12: El anillo **Z** de los números enteros es euclideo.

Demostración: Sea $f : Z - \{0\} \to N$ definida por $f(x) = |x|$ y sean $a, b \in Z - \{0\}$.

Para que **Z** sea un anillo euclideo hay que demostrar que para cualquier $a, b \in Z - \{0\}$ existen $q, r \in Z$ tales que

$$a = b \cdot q + r \tag{5.2}$$

, donde

$$r = 0 \ \ ó \ \ |r| < |b| \tag{5.3}$$

Según el teorema 4.4, existirán $q, r \in N \subset Z$ tales que

$$|a| = |b| \cdot q + r \qquad (r = 0 \ \ ó \ \ r < |b|). \tag{5.4}$$

A continuación, cuatro casos son posibles:

1) Si $a, b \in Z_+^*$, de (5.4) resulta que $a = b \cdot q + r$, donde $q, r \in N \subset Z, \ r = 0 \ \ ó \ \ |r| = r < |b|$.

2) Si $a, b \in Z_-^*$, entonces de (5.4) resulta que $-a = (-b) \cdot q + r \ \Leftrightarrow \ a = b \cdot q + (-r)$, donde

$$q, r \in N \subset Z, \ r = 0 \ \ ó \ \ |-r| = r < -b = |b|.$$

3) Si $a \in Z_+^* \ \ y \ \ b \in Z_-^*$, entonces de (5.4) resulta que $a = (-b) \cdot q + r$, donde

$$q, r \in N \subset Z, \ r = 0 \ \ ó \ \ |r| = r < -b = |b|.$$

4) Si $a \in Z_-^* \ \ y \ \ b \in Z_+^*$, entonces de (5.4) resulta que $-a = b \cdot q + r \ \Leftrightarrow \ a = b \cdot (-q) - r$, donde

$$q, r \in N \subset Z, r = 0 \ \ ó \ \ |-r| = r < b = |b|.$$

Por tanto, Z es un anillo euclídeo.

Observación 5.1: En **Z** cualquier dos elementos a y b tienen máximo común divisor y mínimo común múltiplo. Al ser 1 y -1 los únicos elementos invertibles de Z, si d es un máximo común divisor de a y b entonces $-d$ lo es también. Igualmente, junto a m, $-m$ es también un mínimo común múltiplo de a y b.

Teorema 5.13: Si A es un anillo euclideo, $a_1, b_1 \in A$ y d es un máximo común divisor de a_1 y b_1, entonces existen $x, y \in A$ tales que

$$d = a_1 \cdot x + b_1 \cdot y \tag{5.5}$$

Demostración: Si $a_1 = b_1 = 0$ entonces $d = 0$ y $d = a_1 \cdot 0 + b_1 \cdot 0$.

Si $a_1 \neq 0$ y $b_1 = 0$ entonces $d = a_1 \cdot u_1$, donde u_1 es un elemento invertible y así,

$$d = a_1 \cdot u_1 + b_1 \cdot 0$$

De manera análoga, cuando $a_1 = 0$ y $b_1 \neq 0$, entonces

$$d = a_1 \cdot 0 + b_1 \cdot u_2$$

Luego si $a_1 \neq 0$ y $b_1 \neq 0$, d es un máximo común divisor de a_1 y b_1, $a_1 = a \cdot d$ y $b_1 = b \cdot d$, entonces a y b son primos entre si. Determinar x e y verificando la relación (5.5) es equivalente a determinar x e y verificando la relación

$$e = a \cdot x + b \cdot y \tag{5.6}$$

, donde $a \neq 0$ y $b \neq 0$. Aplicando para a y b el algoritmo de Euclides, se obtienen las relaciones (5.1), donde r_n es un elemento invertible. Así, utilizando las relaciones (5.1) se llega a las equivalencias siguientes:

$$\begin{aligned}
& b \cdot x_1 + r_1 \cdot x = e, \ \text{donde} \ x_1 = q_1 \cdot x + y \\
\Leftrightarrow \ & r_1 \cdot x_2 + r_2 \cdot x_1 = e, \ \text{donde} \ x_2 = q_2 \cdot x_1 + x \\
\Leftrightarrow \ & r_2 \cdot x_3 + r_3 \cdot x_2 = e, \ \text{donde} \ x_3 = q_3 \cdot x_2 + x_1 \\
& \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\Leftrightarrow \ & r_{n-2} \cdot x_{n-1} + r_{n-1} \cdot x_{n-2} = e, \ \text{donde} \ x_{n-1} = q_{n-1} \cdot x_{n-2} + x_{n-3} \\
\Leftrightarrow \ & r_{n-1} \cdot x_n + r_n \cdot x_{n-1} = e, \ \text{donde} \ x_n = q_n \cdot x_{n-1} + x_{n-2}.
\end{aligned}$$

Teniendo en cuenta que r_n es invertible y dando para x_{n+1} un valor cualquiera de A, se obtiene que

$$x_{n-1} = r_n^{-1} \cdot (e - r_{n-1} \cdot x_n) \ \ \text{y} \ \ x_{n-2} = x_n - q_n \cdot x_{n-1}.$$

Al ser ya determinados x_{n-1} y x_{n-2}, resulta que

$$x_{n-3} = x_{n-1} - q_{n-1} \cdot x_{n-2}.$$

Luego, de los valores de x_{n-2} y x_{n-3} se deduce el valor de x_{n-4} y, continuando de esta manera, al final se obtiene que:

$$x = x_2 - q_2 \cdot x_1 \quad e \quad y = x_1 - q_1 \cdot x$$

Observación 5.2: La demostración del teorema 5.13 es constructiva, ya que permite el cálculo efectivo de x e y.

Teorema 5.14: Si A es un anillo integro que no es un cuerpo, el anillo $A[X]$ no es euclideo.
Demostración: Supongamos que $A[X]$ es euclídeo y que $a \in A$ es un elemento no invertible. Obviamente, $a, X \in A[X]$ y sea $d(X)$ un divisor común cualquiera de a y X. Entonces

$$a = d(X) \cdot b(X) \quad , \quad X = d(X) \cdot c(X). \tag{5.7}$$

La primera de las igualdades (5.7) es posible solamente si $d(X)$ y $b(X)$ son polinomios de grado cero, es decir, cuando $d(X) = d_0$ y $b(X) = b_0$, donde $b, d \in A$. Si

$$c(X) = c_0 + c_1 X + \cdots + c_k X^k$$

, entonces

$$X = d_0 \cdot c(X) = d_0 c_0 + d_0 c_1 X + \cdots + d_0 c_k X^k$$

, y así tendremos

$$dc_1 = 1 \;\; y \;\; dc_i = 0 \;\; (i \neq 1)$$

Por tanto d y c_1 son elementos invertibles en el anillo A Puesto que d era un divisor común cualquiera de a y X, resulta que a y X no tienen otros divisores comunes que los elementos invertibles del anillo A. Así, el elemento neutro e del anillo es un máximo común divisor de a y X y, según el teorema 5.13, existirán dos polinomios $p(X)$ y $q(X)$ de $A[X]$ tales

$$e = a \cdot p(X) + X \cdot q(X) \tag{5.8}$$

Suponiendo que

$$p(X) = p_0 + p_1 X + \cdots + p_n X^n$$

, de (5.8) se obtiene que

$$e = ap_0 + X \cdot q'(X)$$

, de donde resulta que

$$ap_0 = e$$

, es decir, a es un elemento invertible de A. La contradicción obtenida demuestra que $A[X]$ no puede ser un anillo euclídeo.
Consecuencia 1: Si en el teorema 5.14 se considera $A = Z$, resulta que $Z[X]$ no es un anillo euclideo.
Teorema 5.15: Si el anillo integro A no es un cuerpo entonces el anillo $A[X]$ de los polinomios con coeficientes en A, no es euclídeo.
Cosidrando el conjunto de los enteros de Gauss:

$$Z[i] = \{ a + bi \;\; / \; a, b \in Z \}$$

, es fácil verificar que $Z[i]$ es un anillo con respecto a la adición y la multiplicación de los números complejos. Es también integro puesto que si $z_1 = a_1 + ib_1$ y $z_2 = a_2 + ib_2$ entonces de

$$z_1 \cdot z_2 = a_1 a_2 - b_1 b_2 + i\,(a_1 b_2 + a_2 b_1) = 0 \;\; y \;\; z_1 \neq 0$$

, resulta que el siguiente sistema homogéneo con las incógnitas $a_2 \; y \; b_2$

$$a_1 a_2 - b_1 b_2 = 0$$
$$b_1 a_2 + a_1 b_2 = 0$$

, tiene el determinante $a_1^2 + a_2^2 \neq 0$, es decir su única solución es $a_2 = 0, b_2 = 0$, que es equivalente a $z_2 = 0$.
Por otra parte, $x + iy \in Z[i]$ es invertible en $Z[i]$ si y solamente si existe $u + iv \in Z[i]$ tal que

$$(x + iy) \cdot (u + iv) = 1 \quad \Leftrightarrow \quad \begin{cases} xu - yv = 1 \\ yu + xv = 0 \end{cases} \quad \Leftrightarrow \quad u = \frac{x}{x^2 + y^2} \quad y \quad v = \frac{-y}{x^2 + y^y}$$

Puesto que $u \; y \; v$ son enteros solamente si $x = 0 \;\; e \;\; y^2 = 1$ ó $x^2 = 1 \;\; e \;\; y = 0$, resulta que en $Z[i]$ existen solo los siguientes elementos invertibles (unidades): 1,-1, i, -i. Así $Z[i]$ no es un cuerpo y por tanto el anillo $Z[i][X]$ no es euclideo.
Teorema 5.16: El anillo $Z[i]$ es euclideo.
Demostración: Si $f : Z[i] - \{0\} \to N$ es la aplicación definida por $f(a + bi) = a^2 + b^2$
, entonces

$$f[(a + bi)(c + di)] = f(a + bi)\, f(c + di). \tag{5.9}$$

, puesto que

$$f[(a+bi)(c+di)] = f[(ac-bd)+(ad+bc)i] = (ac-bd)^2 + (ad+bc)^2 =$$
$$= (a^2+b^2)(c^2+d^2) = f(a+bi)\,f(c+di).$$

Por tanto, si $u_1+u_2i,\ v_1+v_2i \in Z[i]-(0)$ y v_1+v_2i es un divisor de u_1+u_2i entonces existe $q_1+q_2i \in Z[i]$ tal que $u_1+u_2i = (v_1+v_2i)(q_1+q_2i)$, de donde resulta que

$$f(u_1+u_2i) = f[(v_1+v_2i)(q_1+q_2i)] = f(v_1+v_2i)((q_1^2+q_2^2) \geq f(v_1+v_2i)$$

, puesto que $q_1, q_2 \in Z^*$ y $q_1^2+q_2^2 \geq 1$.

Por otra parte, sea

$$\frac{a+bi}{c+di} = u+vi$$

, donde

$$u = \frac{ac+bd}{c^2+d^2} \quad y \quad v = \frac{bc-ad}{c^2+d^2}$$

Si t y s son los enteros más próximos a los números reales u y v (respectivamente) y $f(a+bi) \geq f(c+di)$, entonces $s^2+t^2 \neq 0$ y

$$a+bi = (c+di)[t+si+(u-t)+(v-s)i]$$

, es decir

$$a+bi = (c+di)(t+si)+(c+di)[(u-t)+(v-s)i].$$

Puesto que $a+bi$ y $(c+di)(t+si)$ pertenecen a $Z[i]$ resulta que

$$(c+di)[(u-t)+(v-s)i = a+bi-(c+di)((t+si)] \in Z(i)$$

Introduciendo las notaciones:

$$q' = t\ ,\ q'' = s \quad y \quad r'+r''i = a+bi-(c+di)(t+si)$$

, se puede escribir que

$$a+bi = (c+di)(q'+q''i)+r'+r''i \quad q',q'',r',r'' \in Z \tag{5.10}$$

, y según la igualdad (5.9),

$$f(r'+ir'') = f\{(c+di)[(u-t)+(v-s)i]\} = f(c+di)\,f[(u-t)+(v-s)i] = \tag{5.11}$$
$$= f(c+di)[(u-t)^2+(v-s)^2] < f(c+di)[(1/2)^2+(1/2)^2] = (1/2)\,f(c+di) < f(c+di)$$

En los anillos euclideos, el máximo común divisor se halla por el algoritmo de Euclid que consiste en una serie de divisiones sucesivas. En el caso de los números a_1+b_1i e c_1+d_1i esta serie de divisiones es la siguiente:

$$a_1+b_1i = (c_1+d_1i).(q_1^1+q_2^1i)+r_1^1+r_2^1i$$
$$c_1+d_1i = (r_1^1+r_2^1i).(q_1^2+q_2^2i)+r_1^2+r_2^2i$$
$$r_1^1+r_2^1i = (r_1^2+r_2^2i).(q_1^3+q_2^3i)+r_1^3+r_2^3i$$
$$- -$$
$$r_1^k+r_2^ki = (r_1^{k+1}+r_2^{k+1}i).(q_1^{k+2}+q_2^{k+2}i)+r_1^{k+2}+r_2^{k+2}i$$
$$- -$$
$$r_1^{\acute{}n-2}+r_2^{n-2}i = (r_1^{n-1}+r_2^{n-1}i).(q_1^n+q_2^ni)+r_1^n+r_2^ni$$
$$r_1^{\acute{}n-1}+r_2^{n-1}i = (r_1^n+r_2^ni).(q_1^{n+1}+q_2^{n+1}i)$$

, donde

$$f(a_1+b_1i) \geq f \geq f(c_1+d_1i)$$
$$f(c_1+d_1i) > f(r_1^1+r_2^1i) > \quad \cdots \quad f(r_1^k+r_2^ki) > \quad \cdots \tag{5.11.b}$$

Puesto que la sucesión (5.11.b) es estrictamente decreciente y existe solamente un número finito de números naturales menores que $f(c_1+d_1i)$, tiene que existir un índice n tal que $f(r_1^{n+1}+r_2^{n+1}i) = 0$. Así el máximo común divisor a_1+b_1i e c_1+d_1i será $r_1^n+r_2^ni$. De lo anteriormente expuesto resulta que $Z[i]$ es euclideo. De esta manera, en el anillo $Z[i]$ se puede hablar del máximo común divisor y del mínimo común múltiplo de dos o varios elementos.

Por ejemplo, el máximo común divisor de los enteros de Gauss $11-3i$ y $4+6i$ se puede hallar de la manera siguiente:

$$\frac{11-3i}{4+6i}=\frac{1}{2}-\frac{3}{2}i \quad \Rightarrow \quad q'=t=0 \ y \ q''=s=-1$$
$$r'+r''i=11-3i-(4+6i)(0-i)=5+i$$

Así,

$$11-3i=(4+6i)(-i)+5+i$$

Luego,

$$\frac{4+6i}{5+i}=1+i$$
$$4+6i=(5+i)(1+i)+0$$

Un máximo común divisor es el último resto no nulo en el algoritmo de Euclides, es decir es el número $5+i$. Un mínimo común múltiplo de los números anteriores es:

$$\frac{(11-3i)(4+6i)}{5+i}=14+8i$$

El número $(a+bi)\in Z[i]$ es una unidad (un elemento invertible) si

$$\frac{1}{a+bi}=\frac{a-bi}{a^2+b^2}\in Z[i] \quad \Leftrightarrow a^2+b^2=1 \quad \Leftrightarrow \quad (a^2=1 \ y \ b^2=0) \ ó \ (a^2=0 \ y \ b^2=1)$$
$$\Leftrightarrow \quad (a=\pm 1 \ y \ b=0) \ ó \ (a=0 \ y \ b=\pm 1)$$

Por tanto las unidades del anillo $Z[i]$ son $1, -1, i, -i$. Así, en el ejemplo anterior, multiplicando el máximo común divisor $5+i$ con $-1, i \ e -i$ se obtienen otros valores posibles del máximo común divisor, como $-5-i$, $-1+5i$ y $1-5i$. De la misma manera, junto al mínimo común múltiplo 14+8i tenemos también a los valores siguientes: $-14-8i, -8+14i \ y \ 8-14i$

Luego, puesto que los únicos elementos invertibles de $Z[i]$ son 1, -1, i e –i, $Z[i]$ no es un cuerpo.

Cosecuencia 2: Considerando en el teorema 5.14 $A=Z[i]$, resulta que $Z[i][X]$ no es un anillo euclideo.

Para operar y para hallar el máximo común divisor y el mínimo común múltiplo en el anillo $Z[i]$, se pueden utilizar las funciones siguientes:

```
Public Function DivEEG(ByVal z11 As Double, ByVal z12 As Double, ByVal z21 As Double, ByVal z22 As Double) As Variant
        Dim x(2) As Double, q(2) As Double, r() As Double, v(2, 2) As Double, rr() As Double, d As Double
        rr() = DivNC(z11, z12, z21, z22)
        For i = 1 To 2
                x(i) = Fix(rr(i)): d = Abs(x(i) - rr(i))
                If d <= 0.5 Then
                        q(i) = x(i)
                Else
                        If rr(i) >= 0 Then q(i) = x(i) + 1 Else q(i) = x(i) - 1
                End If
        Next i
    r() = MultNC(z21, z22, q(1), q(2))
    r() = ResNC(z11, z12, r(1), r(2))
    v(1, 1) = q(1): v(1, 2) = q(2)
    v(2, 1) = r(1): v(2, 2) = r(2)
    DivEEG = v()
End Function
'-----------------------------------------------------------------------------------------------------
Public Function MCDEG(ByVal z11, ByVal z12 As Double, ByVal z21 As Double, ByVal z22 As Double) As Variant
        Dim u11 As Double, u12 As Double, u21 As Double, u22 As Double
        Dim rr() As Double, r() As Double, res(2) As Double
        Dim zz1 As Double, zz2 As Double
        u11 = z11: u12 = z12: u21 = z21: u22 = z22
        If u11 * u11 + u12 * u12 < u21 * u21 + u22 * u22 Then
                zz1 = u11: zz2 = u12: u11 = u21: u12 = u22: u21 = zz1: u22 = zz2
        End If
        Do
                rr() = DivEEG(u11, u12, u21, u22)
                If rr(2, 1) = 0 And rr(2, 2) = 0 Then Exit Do
                u11 = u21: u12 = u22: u21 = rr(2, 1): u22 = rr(2, 2)
        Loop
        res(1) = u21: res(2) = u22
        MCDEG = res()
End Function
```

```
'----------------------------------------------------------------------------------------------------
Public Function MCMEG(ByRef z11 As Double, ByVal z12 As Double, ByVal z21 As Double, ByVal z22 As Double) As Variant
        Dim prod() As Double, res() As Double, mcd() As Double
        prod() = MultNC(z11, z12, z21, z22)
        mcd() = MCDEG(z11, z12, z21, z22)
        res() = DivNC(prod(1), prod(2), mcd(1), mcd(2))
        MCMEG = res()
End Function
'----------------------------------------------------------------------------------------------------
Public Function DivNC(ByVal u1 As Double, ByVal u2 As Double, ByVal v1 As Double, ByVal v2 As Double) As Variant
        Dim cmv As Double, co() As Double, rr() As Double
        ReDim co(2)
        cmv = v1 * v1 + v2 * v2
        rr() = MultNC(u1, u2, v1, -v2)
        co(1) = rr(1) / cmv: co(2) = rr(2) / cmv
        DivNC = co()
End Function
'----------------------------------------------------------------------------------------------------
Public Function SumNC(ByVal u1 As Double, ByVal u2 As Double, ByVal v1 As Double, ByVal v2 As Double) As Variant
        Dim rr(2) As Double
        rr(1) = u1 + v1: rr(2) = u2 + v2
        SumNC = rr()
End Function
'----------------------------------------------------------------------------------------------------
Public Function ResNC(ByVal u1 As Double, ByVal u2 As Double, ByVal v1 As Double, ByVal v2 As Double) As Variant
        Dim rr(2) As Double
        rr(1) = u1 - v1: rr(2) = u2 - v2
        ResNC = rr()
End Function
'----------------------------------------------------------------------------------------------------
Public Function MultNC(ByVal x1 As Double, ByVal y1 As Double, ByVal x2 As Double, ByVal y2 As Double) As Variant
   ' Multiplicación de los números complejos.
   Dim pc() As Double
   ReDim pc(2)
   pc(1) = x1 * x2 - y1 * y2
   pc(2) = x1 * y2 + x2 * y1
   MultNC = pc()
End Function
'----------------------------------------------------------------------------------------------------
Public Function FormNC(pr As Double, pi As Double) As String
        ' Escritura de un número complejo en una caja de texto.
        Dim r As String
        If pr <> 0 Then r = r + Str$(pr)
        If pi <> 0 Then
                If Abs(pi) = 1 Then
                        If pi = 1 Then r = r + " + "
                        If pi = -1 Then r = r + " - "
                Else
                        If pi > 0 Then
                                If pr <> 0 Then r = r + " + "
                                 r = r + Str$(pi)
                        Else
                                r = r + " - " + Mid$(Str$(pi), 2)
                        End If
                 End If
                 r = r + " i"
        End If
        If r = "" Then r = "0"
        FormNC = r
End Function
```

Construyendo un programa con las funciones anteriores, se obtiene que un máximo común divisor y un mínimo común múltiplo de los enteros de Gauss $1021+1405i$ y $-1131+3533i$ son $-123-83i$ y $26573-34339i$, respectivamente.

Si se desea hacer cálculos con enteros de Gauss cuya parte real o imaginaria tiene más de 15 cifras, se deben utilizar las operaciones con enteros largos y en lugar de las funciones anteriores hay que emplear las siguientes:

```
Public Function DivEEGG(ByVal z11 As String, ByVal z12 As String, ByVal z21 As String, ByVal z22 As String) As Variant
        Dim x(2) As String, q(2) As String, r() As String, v(2, 2) As String, pp As String
        Dim rr() As String, y(2) As String, t As String, i As Integer, k As Integer
        rr() = DivNCG(z11, z12, z21, z22)
        For i = 1 To 2
                y(i) = FixNG(rr(i))
                x(1) = y(i): x(2) = rr(i): x(1) = Restar(x(), 7)
                If Left$(x(1), 1) = "-" Then x(1) = Mid$(x(1), 2)
                x(2) = "0.5": t = RestarDec(x(), 7)
                If t = "0" Or Left$(t, 1) = "-" Then
                        q(i) = y(i)
```

```
                Else
                        x(1) = y(i): x(2) = "1"
                        If Left$(rr(i), 1) = "-" Then
                                q(i) = Restar(x(), 7)
                        Else
                                q(i) = Sumar(x(), 7)
                        End If
                End If
        Next i
        r() = MultNCG(z21, z22, q(1), q(2))
        r() = ResNCG(z11, z12, r(1), r(2))
        v(1, 1) = q(1): v(1, 2) = q(2)
        v(2, 1) = r(1): v(2, 2) = r(2)
        DivEEGG = v()
End Function
' - - -
```

```
        x(1) = u1: x(2) = v2: p1 = Multiplicar(x(), 7)
        x(1) = u2: x(2) = v1: p2 = Multiplicar(x(), 7)
        x(1) = p1: x(2) = p2: pc(2) = Sumar(x(), 7)
        MultNCG = pc()
End Function
' - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Public Function FixNG(ByVal u As String) As String
        Dim v As String, pp As String, k As Integer
        If u = "-1" Then FixNG = "-1": Exit Function
        If u = "1" Then FixNG = "1": Exit Function
        If u = "0" Then FixNG = "0": Exit Function
        If Left$(u, 2) = "0." Then FixNG = "0": Exit Function
        If Left$(u, 3) = "-0." Then FixNG = "-1": Exit Function
        For k = 1 To Len(u)
                pp = Right$(Left$(u, k), 1)
                If pp = "." Then
                        v = Left$(u, k - 1): Exit For
                End If
                If pp <> "." Then v = u
        Next k
         FixNG = v
End Function
' - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Public Function FormNCG(pr As String, pi As String) As String
        'Escritura de un número complejo en una caja de texto.
        Dim r As String
        If pr <> "0" Then
                r = r + pr
        End If
        If pi <> "0" Then
                If pi = "1" Or pi = "-1" Then
                        If pi = "1" Then r = r + " + "
                        If pi = "-1" Then r = r + " - "
                Else
                        If Left$(pi, 1) <> "-" Then
                                If pr <> "0" Then r = r + " + " + pi
                        Else
                                r = r + " - " + Mid$(pi, 2)
                        End If
                End If
                r = r + " i"
         End If
         If r = "" Then r = "0"
        FormNCG = r
End Function
```

Aplicando el código anterior, a los enteros de Gauss:

$$z_1 = 5534445886\,54544 + 7619867888 \quad y \quad z_2 = 5439668886\,887 + 987646584\,i$$

, se obtiene que $(z_1, z_2) = 1 + 2i$ e

$$[z_1, z_2] = 7111791668\,8624828726\,5318928 - 1421671211\,8984406086\,037375504\,i$$

Teorema 5.17: Si en un anillo euclideo A, a es divisor de $b \cdot c$ y a y b son primos relativos, entonces a es un divisor de c.

Demostración: Al ser a y b primos relativos, e es un máximo común divisor de ellos y, por tanto, existirán $x, y \in A$ tales que

$$a \cdot x + b \cdot y = e$$

, de donde resulta que

$$a \cdot (x \cdot c) + (b \cdot c)y) = e \qquad (5.12)$$

Dado que $b \cdot c = a \cdot q$, según (5.12)

$$a \cdot (x \cdot c + q \cdot y) = c$$

, y así a es divisor de c.

Definición 5.5: Se dice que $a \in A - \{0\}$ es un elemento irreducible en el anillo $(A,+,\cdot)$ si es un elemento irreducible en el semigrupo $(A - \{0\},\cdot)$.

Definición 5.6: $a \in A - \{0\}$ es un elemento primo en el anillo $(A,+,\cdot)$ si es un elemento primo del semigrupo $(A - \{0\},\cdot)$.

Teniendo en cuenta que en un anillo euclideo dos elementos cualquiera tienen máximo común divisor, según

los teoremas (3.13) y (3.15), un elemento no invertible y no nulo es irreducible si, y solamente si, es un elemento primo.

Teorema 5.18: Si A es un anillo conmutativo con elemento neutro e, $f(X)\in A[X]$ y $f(X)$ no es el polinomio nulo, entonces $f(X)$ es invertible en $A[X]$ sí y solamente si $f(X)$ es un polinomio de grado nulo.

Demostración: Si $f(X)$ es invertible en $A[X]$, existirá $g(X)\in A[X]$ tal que

$$f(X)\cdot g(X)) = e$$

, de donde resulta que

$$grado\left(f(X)\right)+grado\left(g(X)\right)=0$$

, y así tendremos también:

$$grado\left(f(X)\right)=0 \;\; y \;\; grado\left(g(X)\right)=0.$$

Por tanto,

$$f(X)=a\,,\; g(X)=a' \;\; y \; a\cdot a'=e$$

Así, $f(X)$ es invertible en A.

Al revés, si $f(X)=a$ es un elemento invertible en A, entonces $g(X)=a^{-1}$ es el elemento inverso de $f(X)$ en $A[X]$.

Si K es un cuerpo conmutativo, entonces el conjunto de los elementos invertibles del anillo $K[X]$ es $K-\{0\}$.

Teorema 5.19: Si K es un cuerpo conmutativo entonces $K[X]$ es un anillo euclídeo.

Demostración: Si $\theta(X)$es el polinomio nulo de $K[X]$, sea

$$\psi : K[X]-\{\theta(X)\}\to N$$

, la aplicación definida por

$$\psi\left[f(X)\right]=grado\left(f(X)\right),\text{ cualquiera que sea } f(X)\in K[X]-\{\theta(X)\}$$

Si $f(X), g(X)\in K[X]$, $g(X)\neq\theta(X)$ y $grado\left(g(X)\right)=0$, entonces $g(X)\in K-\{0_K\}$ y

$$f(X)=g(X)\cdot\left[\left(g(X)^{-1}\cdot f(X)\right)\right]$$

Por tanto,

$$f(X)=g(X)\cdot q(X)+r(X)$$

, donde

$$q(X)=\left(g(X)\right)^{-1}\cdot f(X) \;\; y \;\; r(X)=\theta(X)$$

Si, $grado\left(g(X)\right)\neq 0$, la demostración se hará por inducción completa con respecto al grado del polinomio $f(X)$. Si $grado\left(f(X)\right)=0$, entonces $f(X)\in K$ y

$$f(X)=g(X)\cdot\theta(X)+f(X)$$

, donde

$$grado\left(f(X)\right)=0<grado\left(g(X)\right)\neq 0$$

Sean ahora

$$f(X)=a_0+a_1X+\cdots+a_nX^n \quad \text{y} \quad g(X)=b_0+b_1X+\cdots+a_mX^m$$

Suponiendo que el teorema es verdadero para todos los polinomios $f(X)$ de grado menor que n. A continuación dos casos son posibles.

1) Si $m>n$, entonces

$$f(X)=g(X)\cdot\theta(X)+r(X),\text{ donde, } grado\;\left(f(X)\right)=n<m=grado\;\left(g(X)\right)$$

2) Si $m\leq n$, el polinomio

$$h(X)=f(X)-a_nb_m^{-1}X^{n-m}\cdot g(X)$$

, es de grado menor que n ya que los polinomios

$$f(X) \;\; y \;\; a_nb_m^{-1}X^{n-m}\cdot g(X)$$

, tienen el mismo monomio de más alto grado, igual con a_nX^n. Ahora, según la hipótesis de la inducción , el teorema se cumple para el polinomio $h(X)$, es decir

$$h(X) = g(X)\cdot q_1(X) + r(X), \text{ donde } \quad grado\ (r(X)) < grado\ (g(X))$$

Así,

$$f(X) - a_n b_m^{-1} X^{n-m}\cdot g(X) = g(X)\cdot q_1(X) + r(X)$$

, es decir

$$f(X) = g(X)\cdot q(X) + r(X)$$

, donde

$$q(X) = q_1(X) + a_n b_m^{-1} X^{n-m} \quad y \quad grado\ (r(X)) < grado\ (g(X))$$

Así pues, si el teorema se supone verdadero para los polinomios de grado menor que n, según 1) y 2) resulta que será verdadero también para un polinomio de grado n y entonces, según el principio de la inducción completa, será verdadero para cualquier polinomio $f(X)$.

Si Q, R y C designan los cuerpos de los números racionales reales y complejos, respectivamente, entonces $Q[X]$, $R[X]$ y $C[X]$ son anillos euclídeos y, por consiguiente, en estos anillos existe el máximo común divisor y el mínimo común múltiplo de dos polinomios cualesquiera. Si

$$f(X) = a_n X^n + a_{n-1} X^{n-1} + \cdots + a_1 X + a_0, \quad g(X) = b_m X^m + b_{m-1} X^{m-1} + \cdots + b_1 X + b_0 \quad (n \geq m)$$

, el cálculo práctico de los polinomios $q(X)$ y $r(X)$ en la división euclidea $f(X) = g(X)\cdot q(X) + r(X)$ se efectua mediante el conocido algoritmo de Euclides. Obviamente,

$$f_1(X) = f(X) - a_n b_m^{-1} X^{n-m}\cdot g(X) \quad \Rightarrow \quad grado\,(f(X)) > grado(f_1(X)).$$

Si $grado\ (f_1(X)) < m$, entonces paramos. En el caso contrario se considera

$$f_1(X) = a_{n_1 1} X^{n_1} + \cdots + a_{11} X + a_{01} \quad \text{, y } \quad f_2(X) = f_1(X) - a_{n_1 1} b_m^{-1} X^{n_1 - m}\cdot g(X), \text{ donde}$$

$$grado\ (f_1(X)) > grado\ (f_2(X))$$

Si $grado\ (f_2(X)) < m$, entonces se para. En el caso contrario se pone

$$f_2(X) = a_{n_2 2} X^{n_2} + \cdots + a_{12} X + a_{02}, \text{ y } \quad f_3(X) = f_2(X) - a_{n_2 2} b_m^{-1} X^{n_2 - m}\cdot g(X), \text{donde}$$

$$grado(f_2(X)) > grado(f_3(X))$$

Si $grado\,(f_3(X)) < m$, entonces se para, mientras que en el caso contrario se continua el proceso anterior.

La sucesión,

$$grado\ (f(X)),\ grado\ (f_1(X)),\ grado\ (f_2(X)), \ldots$$

, es una sucesión de números naturales estrictamente decreciente y, por tanto, debe ser una sucesión finita. Así, en el proceso anterior debe existir un índice k tal que

$$f_k(X) = a_{n_k 1} X^{n_k} + \cdots + a_{1k} X + a_{0k}$$

$$f_{k+1}(X) = f_k(X) - a_{n_k k} b_m^{-1} X^{n_k - m}\cdot g(X)$$

, donde

$$grado(f_{k+1}(X)) > grado(f_k(X)) \quad y \quad grado(f_{k+1}(X)) < m.$$

Entonces

$$f(X) = a_n b_m^{-1} X^{n-m}\cdot g(X) + f_1(X)$$

$$= \left[a_n b_m^{-1} X^{n-m} + a_{n_1 1} b_m^{-1} X^{n_1 - m}\right]\cdot g(X) + f_2(X)$$

.....

$$= \left[a_n b_m^{-1} X^{n-m} + a_{n_1 1} b_m^{-1} X^{n_1 - m} + \cdots + a_{n_k k} b_m^{-1}\right]\cdot g(X) + f_{k+1}(X)$$

$$= q(X)\cdot g(X) + r(X)$$

, donde

$$q(X) = a_n b_m^{-1} X^{n-m} + a_{n_1 1} b_m^{-1} X^{n_1 - m} + \cdots + a_{n_k k} b_m^{-1} X^{n_k - m} \quad y \quad r(X) = f_{k+1}(X).$$

Ejemplo 5.1: Si

$$f(X)=X^3-4X^2+5X+2 \;\; y \;\; g(X)=2X^2-X+1$$

, entonces en $Q[X]$ la división euclidea $f(X)=q(X)\cdot g(X)+r(X)$ se puede efectuar de la manera siguiente:

X^3	$-4X^2$	$+5X$	$+2$	$2X^2$	$-X$	$+1$
$-X^3$	$1/2X^2$	$-1/2X$		$1/2X$	$-7/4$	
0	$-7/2X^2$	$9/2X$	$+2$	$+2$		
	$7/2X^2$	$-7/4X$	$+7/4$			
	0	$11/4X$	$+15/4$			

Así

$$q(X)=\frac{1}{2}X-\frac{7}{4} \;\; y \;\; r(X)=.\frac{11}{4}X+\frac{15}{4}$$

Se dice que $-\frac{7}{2}X^2+\frac{9}{2}X+2$ es un resto parcial (intermedio) de la división y que $\frac{11}{4}X+\frac{15}{4}$ es el resto.

Observación 5.3: Si A es un anillo euclideo con el elemento neutro e y $CFr(A)$ es su cuerpo de fracciones, en una división euclidea en el anillo $CFr(A)[X]$, cuándo el dividiendo y el divisor (de grado n y m, respectivamente) tienen coeficientes pertenecientes al anillo A , al multiplicar el dividiendo y los restos parciales con los números $\lambda_0,\lambda_1,\ldots\lambda_p\in A$ $(p=n-m)$, se obtiene el "cociente" $q^*(X)$ y el "resto" $r^*(X)$. Si $q(X)$ y $r(X)$ son el cociente y el resto verdaderos en $CFr(A)[X]$,

$$q*(X)=c_0X^p+\cdots+c_{p-1}X+c_p \;\;,\;\; q(X)=d_0X^p+\cdots+d_{p-1}X+d_p$$

, entonces

$$d_i=c_i/q_i \;\;,\;\; q_i=\lambda_0.\cdots\lambda_i \;\; \text{y} \;\; r(X)=\frac{r^*(X)}{\lambda_0\cdot\lambda_1\cdots\lambda_p} \qquad (5.13)$$

En efecto, al multiplicar el resto parcial $f_k(X)$ con el número entero λ, de

$$f_{k+1}(X)=f_k(X)-a_{n_k k}-b_m^{-1}X^{n_k-m}\cdot g(X)$$

, resulta que $f_{k+1}(X)$ se multiplicará también por λ ya que $f_k(X)$ y $a_{n_k k}$ han sido multiplicados por λ. Por tanto, todos los restos parciales posteriores a $f_k(X)$ serán multiplicados por λ. Obviamente, se multiplicarán por λ todos aquellos términos del cociente que corresponden al resto parcial $f_k(X)$ o a restos parciales posteriores a esto.

Normalmente los factores enteros $\lambda_0,\lambda_1,\ldots,\lambda_p$ se eligen de la manera siguiente:

$$\lambda_0\cdot|a_n|=mcm\left(|a_n|,|b_m|\right)$$
$$\lambda_k\cdot\left|a_{n_k k}\right|=mcm\left(\left|a_{n_k}k\right|,|b_m|\right) \qquad (k=1,\ldots,p)$$

, para evitar los cálculos con fracciones.

Hay que mencionar que algunos de los números λ_i pueden tener el valor e cuando el polinomio ó el resto parcial correspondiente no se multiplico con ningún número.

Observación 5.4: Si A es un anillo euclideo y $CFr(A)$ es su cuerpo de fracciones, cualesquiera que sean los polinomios $f(X)$ y $g(X)$ con coeficientes en el anillo A, existen $\lambda,\mu\in A$ y $q(X),r(X)\in A[X]$ tal que

$$\lambda\cdot f(X)=g(X)\cdot q(X)+\mu\cdot r(X) \qquad grado\left(r(X)\right)<grado\left(g(X)\right) \qquad (5.14)$$

En efecto, de la observación 5.3.a resulta que

$$\lambda_0\cdot\lambda_1\cdot\cdots\cdot\lambda_p f(X)=g(X)\cdot q^{**}(X)+\mu\cdot r^{**}(X) \;\; grado\left(r^{**}(X)\right)<grado\left(g(X)\right)$$

, donde μ es el máximo común divisor de los coeficientes de $r^*(X)$, $r^{**}(X)=\frac{r^*(X)}{\mu}$ y

$$q^{**}(X)=h_0X^p+\cdots+h_{p-1}X+h_p \quad (h_i=c_i\cdot\lambda_{i+1}\cdots\lambda_p).$$

Así $\lambda=\lambda_0\lambda_1\cdots\lambda_n$, $q(X)=q^{**}(X)$ y $r(X)=r^{**}(X)$.

De la igualdad (5.14) resulta que cualquier divisor común de los polinomios $f(X)$ y $g(X)$ es un divisor común de $g(X)$ y $r(X)$ y al revés.

Ejemplo 5.2: La observación 5.3 se puede comprobar en un caso particular, de la manera siguiente. Sean

$$f_1(X) = 2X^4 + 3X^3 + 4X^2 + X - 2 \;\; y \;\; f_2(X) = 3X^3 + 2X^2 + 5X - 2$$

Entonces, $A = Z \;\; AF = Q$ y en $Q[X]$ la división de $f_1(X)$ entre $f_2(X)$ se hace según la tabla siguiente:

$2X^4$	$+\ 3X^3$	$+\ 4X^2$	$+\ X$	-2	$3X^3$	$+2X^2$	$+5X$	-2
$-2X^4$	$-(4/3)X^3$	$-(10/3)X^2$	$+(4/3)X$		$(2/3)X$	$+5/9$		
0	$(5/3)X^3$	$+ (2/3(X^2$	$+ (7/3)X$	-2				
	$-(5/3)X^3$	$-(10/9)X^2$	$-(25/9)X$	$10/9$				
	0	$- (4/9)X^2$	$- (4/9)X$	$-8/9$				

Para evitar el cálculo con fracciones se puede proceder también de la manera siguiente:

$2X^4$	$+3X^3$	$+4X^2$	$+ X$	-2	$3X^3$	$+2X^2$	$+5x$	-2
$6X^4$	$+9X^3$	$-12X^2$	$+3X$	-6	$2\ X$	$+5$		
$-6X^4$	$-4X^3$	$-10x^2$	$+4X$					
0	$5X^3$	$+ 2X^2$	$+7X$	-6				
	$15X^3$	$+ 6X^2$	$+21X$	-18				
	$-15X^3$	$-10X^2$	$-25X$	$+10$				
	0	$-4X^2$	$-4X$	-8				

, donde $\lambda_0 = 3 \; y \; \lambda_1 = 3$, $q^*(X) = 2X + 5$ y $r^*(X) = -4X^2 - 4X - 8$. Luego,

$$q(X) = (2/\lambda_0)X + 5/(\lambda_0 \cdot \lambda_1) = \frac{2}{3}X + \frac{5}{9}$$

$$r(X) = \frac{r^*(X)}{\lambda_0 \cdot \lambda_1} = \frac{-4X^2 - 4X - 8}{9} = -\frac{4}{9}X^2 - \frac{4}{9}X - \frac{8}{9}$$

El código siguiente efectúa los cálculos de arriba, es decir realiza la división de dos polinomios con coeficientes enteros según la observación 5.3 y devuelve el cociente y el resto (en general con coeficientes fraccionarios).

```
Public Function DivisionPolinomios(ByRef c10() As String, ByRef c20() As String) As Variant
        ‘ Los coeficientes deben ser enteros.
        Dim i As Integer, j As Integer, k As Integer, x(2) As String, mx As String
        Dim g1 As Integer, g2 As Integer, gc As Integer, gr As Integer, mp As String
        Dim ym As String, m1() As String, m2 As String, mc() As String, n As Integer
        Dim xx() As String, ya As String, rr() As String, rr0 As String, cqr(2) As String
        Dim c1() As String, c2() As String
        c1() = c10(): c2() = c20()
        n = 7: g1 = UBound(c1()): g2 = UBound(c2()):
        If g1 < g2 Then
                cqr(1) = "0"
                cqr(2) = FormatoDividendoDivisor(c2())
                DivisionPolinomios = cqr(): Exit Function
        End If
        gc = g1 - g2: gr = g2 - 1
```

```
        ReDim m1(gc), xx(gc), mc(gc), y1(gc), y2(gc), z(gr), z1(gr), z2(gr)
        mx = "1"
        If Left$(c2(0), 1) = "-" Then ym = Mid$(c2(0), 2) Else ym = c2(0)
        For i = 0 To g1 - g2
                If c1(i) = "0" Then
                        m1(i) = "1": xx(i) = "0": mc(i) = mx
                Else
                        x(1) = c1(i): ya = x(1): x(2) = ym: mp = MinComMult(x(), n)
                        x(1) = mp: x(2) = ya: rr() = DivisionEuclidea(x(), n): m1(i) = rr(1)
                        If m1(i) <> "1" Then
                                x(1) = mx: x(2) = m1(i): mx = Multiplicar(x(), n)
                                For k = i To g1
                                        x(1) = c1(k): x(2) = m1(i): c1(k) = Multiplicar(x(), n)
                                Next k
                        End If
                        x(1) = c1(i): x(2) = c2(0): rr() = DivisionEuclidea(x(), n)
                        m2 = rr(1): xx(i) = m2
                        For k = i To i + g2
                                x(1) = c2(k - i): x(2) = m2: rp = Multiplicar(x(), n)
                                x(1) = c1(k): x(2) = rp: c1(k) = Restar(x(), n)
                        Next k
                        mc(i) = mx
                End If
        Next i
        j = g1 - g2 + 1
        For i = j To g1: z(i - j) = c1(i): Next i
                For i = 0 To g1 - g2
                        If xx(i) <> "0" Then
                                x(1) = xx(i): x(2) = mc(i): rr0 = MaxComDiv(x(), n)
                                If rr0 <> "1" Then
                                        x(1) = xx(i): x(2) = rr0: rr() = DivisionEuclidea(x(), n): y1(i) = rr(1)
                                         x(1) = mc(i): x(2) = rr0: rr() = DivisionEuclidea(x(), n): y2(i) = rr(1)
                                Else
                                        y1(i) = xx(i): y2(i) = mc(i)
                                End If
                        Else
                                y1(i) = "0": y2(i) = mc(i)
                 End If
        Next i
        For i = 0 To g2 - 1
                If z(i) = "0" Then
                        z1(i) = "0": z2(i) = mx
                Else
                        x(1) = z(i): x(2) = mx
                        If Left$(x(1), 1) = "-" Then x(1) = Mid$(x(1), 2)
                        rr0 = MaxComDiv(x(), n)
                        If rr0 = "1" Then
                                z1(i) = z(i): z2(i) = mx
                        Else
                                x(1) = z(i): x(2) = rr0: rr() = DivisionEuclidea(x(), n): z1(i) = rr(1)
                                x(1) = mx: x(2) = rr0: rr() = DivisionEuclidea(x(), n): z2(i) = rr(1)
                        End If
                End If
        Next i
        cqr(1) = RutinaFra(y1(), y2())
        cqr(2) = RutinaFra(z1(), z2())
        If cqr(2) = "" Then cqr(2) = "0"
        DivisionPolinomios = cqr()
End Function
' - - -
```

```
                End If
            End If
        Next i
        RutinaFra = FormatoCocienteResto(v1(), v2())
End Function
‘ - - -
```

```
                        Else
                                cx = x0(0)
                        End If
                Else
                        If gx <> 0 Then
                                If x0(0) = "-1" Then cx = "-"
                        Else
                                If x0(0) = "-1" Then
                                        cx = Str$(-1)
                                Else
                                        cx = Mid$(Str$(1), 2)
                                End If
                        End If
                End If
                If gx <> 0 Then
                        If gx = 1 Then
                                cx = cx + " X"
                        Else
                                cx = cx + " X^" + Mid$(Str$(gx), 2)
                        End If
                End If
        Else
                If Left$(x0(i), 1) = "-" Then ax = " - " Else  ax = " + "
                If (x0(i) <> "1" And x0(i) <> "-1") Or i = gx Then
                        If Left$(x0(i), 1) = "-" Then
                                ax = ax + Mid$(x0(i), 2)
                        Else
                                ax = ax + x0(i)
                        End If
                End If
                If gx > 1 Then
                        If i < gx - 1 Then
                                ax = ax + " X^"
                                ax = ax + Mid$(Str$(gx - i), 2)
                        Else
                                If i = gx - 1 Then
                                ax = ax + " X"
                        End If
                End If
        End If
        cx = cx + ax: ax = ""
    End If
 End If
Next i
FormatoDividendoDivisor = cx
End Function
```

Aunque $Z[X]$ no es euclideo, de la observación 5.4 resulta que dos polinomios con coeficientes enteros tienen en $Q[X]=CFr(Z)[X]$ siempre un máximo común divisor con coeficientes enteros. Para hallarlo, el algoritmo de Euclides hay que escribirlo en la forma siguiente:

$$\lambda_0 \cdot f(X)=g(X)\cdot q(X)+\mu_0 \cdot r(X) \qquad grado\,(r(X))<grado\,(g(X))$$
$$\lambda_1 \cdot g(X)=r(X)\cdot q_1(X)+\mu_1 \cdot r_1(X) \qquad grado\,(r_1(X))<grado\,(r(X))$$
$$\lambda_2 \cdot r(X)=r_1(X)\cdot q_2(X)+\mu_2 \cdot r_2(X) \qquad grado\,(r_2(X))<grado\,(r_1(X))$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\lambda_{k+2} \cdot r_k(X)=r_{k'-1}(X)\cdot q_k(X)+\mu_{k+2} \cdot r_k(X) \qquad grado\,(r_k(X))<grado\,(r_{k-1}(X))$$
$$\lambda_{k+3} \cdot r_{k+1}(X)=r_{k'}(X)\cdot q_{k+1}(X)$$

Entonces, $r_k(X)$ es el máximo común divisor (con coeficientes enteros) de los dos polinomios. El código que realiza estos cálculos (en el lenguaje visual-Basic) es la siguiente:

```
Public Function CalculoMCD(ByRef c10() As String, ByRef c20() As String) As Variant
        ' ---- CÁLCULO DE M.C.D. DE LOS POLINOMIOS
        Dim i As Integer, j As Integer, k As Integer, g1 As Integer, g2 As Integer
        Dim s1 As Integer, sw As Integer, jr As Integer, rr() As String, p1() As String
        Dim mp As String, m11 As String, ya As String, ym As String, m2 As String
        Dim n As Integer, z() As String, gz As Integer, x(2) As String
        Dim c1() As String, c2() As String
```

```
	c1() = c10(): c2() = c20()
	n = 7: g1 = UBound(c1()): g2 = UBound(c2())
	ReDim z(g2)
	Do
		gz = 0: s1 = 0
		For i = 0 To g1 - g2
			If c1(i) <> "0" Then
				x(1) = c1(i): x(2) = c2(0): x(1) = MinComMult(x(), n): x(2) = c1(i)
				rr() = DivisionEuclidea(x(), n): m11 = rr(1)
				If Left$(m11, 1) = "-" Then m11 = Mid$(m11, 2)
				If m11 <> "1" Then
					For k = i To g1
						x(1) = c1(k): x(2) = m11: c1(k) = Multiplicar(x(), n)
					Next k
				End If
				x(1) = c1(i): x(2) = c2(0): rr() = DivisionEuclidea(x(), n): m2 = rr(1)
				For k = i To i + g2
					x(1) = c2(k - i): x(2) = m2: x(2) = Multiplicar(x(), n)
					x(1) = c1(k): c1(k) = Restar(x(), n)
				Next k
			End If
		Next i
		jr = g1 - g2 + 1: j = jr
		For i = jr To g1
			If c1(i) = "0" Then
				If s1 = 1 Then z(i - j) = c1(i): gz = gz + 1 Else j = j + 1
			Else
				z(i - j) = c1(i): gz = gz + 1
				If s1 = 0 Then s1 = s1 + 1
			End If
		Next i
		sw = 0
		For i = 0 To gz - 1
			If z(i) <> "0" Then
				ReDim p1(gz - 1)
				For j = 0 To gz - 1: p1(j) = z(j): Next j
				c1() = c2(): c2() = p1()
				g1 = UBound(c1()): g2 = UBound(c2())
				 c2() = Rutina(c2(), n)
				sw = 1: Exit For
			End If
		Next i
		If sw = 0 Then Exit Do
	Loop
	c2() = Rutina(c2(), n)
	CalculoMCD = c2()
End Function
‘ - - -
```

```
                End If
            End If
        Next i
    End If
    Rutina = c2x()
End Function
```

Para hallar el mínimo común múltiplo de dos polinomios, tras haber calculado su máximo común divisor, hay que dividir el producto de los polinomios con el máximo común divisor. La división del producto de dos polinomios con coeficientes enteros con su máximo común divisor (con coeficientes enteros) se efectuará mediante la función *DivEXPR*.

```
Public Function CalculoMCM(c10() As String, c20() As String) As Variant
    Dim mcd() As String, mcm() As String, pp() As String, gpp As Integer
    Dim c1() As String, c2() As String
    c1() = c10(): c2() = c20()
    mcd() = CalculoMCD(c1(), c2())
    pp() = ProductoPolinomios(c10(), c20())
    pp(0) = pp(0): pp(1) = pp(1)
    gpp = UBound(pp())
    If Left$(pp(0), 1) = "-" Then
        For i = 0 To gpp
            If Left$(pp(i), 1) = "-" Then
                pp(i) = Mid$(pp(i), 2)
            Else
                pp(i) = "-" + pp(i)
            End If
        Next i
    End If
    mcm() = DivExPR(pp(), mcd())
    CalculoMCM = mcm()
End Function
‘ - - -
```

```
        End If
        If Left$(u(0), 1) = "-" Then
                For i = 0 To gm
                        If u(i) <> "0" Then
                                If Left$(u(i), 1) = "-" Then u(i) = Mid$(u(i), 2) Else u(i) = "-" + u(i)
                        End If
                Next i
        End If
        DivExPR = u()
End Function
```

A veces tiene interés efectuar la división exacta de dos polinomios con coeficientes enteros y luego dividir el resultado con el máximo común divisor de los coeficientes. Por ejemplo, si $A(X)$ es un polinomio con coeficientes enteros, la ecuación $A(X)=0$ puede tener soluciones multiples y entonces para resolver la ecuación es aconsejable sustituirla con una equivalente $A_1(X)=0$ que tenga las mismas soluciones, pero simples y con los coeficientes más pequeños en valor absoluto. Obviamente, si $A'(X)$ es el polinomio derivado de $A(X)$, $A_1(X)$ se puede obtener dividiedo $A(X)$ con el máximo común divisor $D(X)$ de los polinomios $A(X)$ y $A'(X)$ y luego dividir el resultado con el máximo común divisor de sus coeficientes. Para dividir $A(X)$ entre $D(X)$ y luego dividir el resultado con el máximo común divisor de sus coeficientes se puede utilizar la función siguiente:

```
Public Function DivEspPR(ByRef c10() As String, c20() As String) As Variant
        '----- División exacta del producto de los polinomios por el M.C.D de ellos.
        Dim i As Integer, k As Integer, gpp As Integer, g2 As Integer, gm As Integer
        Dim ya As String, mp As String, rq As String, w As String, qx As String
        Dim m1() As String, u() As String, rr0 As String, sw As Integer
        Dim n As Integer, rr() As String, x(2) As String, c1() As String, c2() As String
        c1() = c10(): c2() = c20()
        n = 7:  g1 = UBound(c1()): g2 = UBound(c2())
        gm = g1 - g2
        ReDim u(gm), m1(gm + 1)
        m1(gm + 1) = "1"
        For i = 0 To gm
                If c1(i) = "0" Then
                        m1(i) = "1": u(i) = "0"
                Else
                        x(1) = c1(i): x(2) = c2(0)
                        x(1) = MinComMult(x(), n): x(2) = c1(i)
                        rr() = DivisionEuclidea(x(), n): qx = rr(1)
                        If Left$(qx, 1) = "-" Then qx = Mid$(qx, 2)
                        m1(i) = qx
                        If m1(i) <> "1" Then
                                For k = i To g1
                                        x(1) = c1(k): x(2) = m1(i): c1(k) = Multiplicar(x(), n)
                                Next k
                        End If
                        x(1) = c1(i): x(2) = c2(0)
                        rr() = DivisionEuclidea(x(), n): u(i) = rr(1)
                        For k = i To i + g2
                                x(1) = c2(k - i): x(2) = u(i)
                                x(2) = Multiplicar(x(), n): x(1) = c1(k)
                                c1(k) = Restar(x(), n)
                        Next k
                End If
        Next i
        For k = 0 To gm
                For i = 0 To k
                        x(1) = u(i): x(2) = m1(k + 1): u(i) = Multiplicar(x(), n)
                Next i
        Next k
        sw = 0: rr0 = u(0)
        For i = 1 To gm
                If u(i) <> "0" Then
                        x(1) = u(i): x(2) = rr0: rr0 = MaxComDiv(x(), n)
                        If rr0 = "1" Then sw = 1: Exit For
                End If
        Next i
        If sw = 0 Then
                For i = 0 To gm
```

```
                    x(1) = u(i): x(2) = rr0
                    If u(i) <> "0" Then
                        x(1) = u(i): x(2) = rr0: rr()  = DivisionEuclidea(x(), n)
                        u(i) = rr(1)
                    End If
                Next i
        End If
        If Left$(u(0), 1) = "-" Then
                For i = 0 To gm
                    If u(i) <> "0" Then
                        If Left$(u(i), 1) = "-" Then u(i) = Mid$(u(i), 2) Else u(i) = "-" + u(i)
                    End If
                Next i
        End If
         DivEspPR = u()
End Function
```

Para hallar en $Q[X]$ el mínimo común múltiplo con coeficientes enteros de dos polinomios (con ceficientes enteros) y cuyos coeficientes sean primos relativos, se puede utilizar la función:

```
Public Function CalculoMCM2(c10() As String, c20() As String) As Variant
        Dim mcd() As String, mcm() As String, pp() As String, gpp As Integer
        Dim c1() As String, c2() As String
        c1() = c10(): c2() = c20()
        mcd() = CalculoMCD(c1(), c2())
        pp() = ProductoPolinomios(c10(), c20())
        pp(0) = pp(0): pp(1) = pp(1)
        gpp = UBound(pp())
        If Left$(pp(0), 1) = "-" Then
                For i = 0 To gpp
                    If Left$(pp(i), 1) = "-" Then
                        pp(i) = Mid$(pp(i), 2)
                    Else
                        pp(i) = "-" + pp(i)
                    End If
                Next i
        End If
        mcm() = DivEspPR(pp(), mcd())
        CalculoMCM2 = mcm()
End Function
```

Las funciones que dan como resultado un polinomio, devuelven el polinomio en una matriz unidimensional, cuyos elementos son los coeficientes del polinomio. Para visualizar este resultado en la escritura normal de un polinomio, hay que transformar la matriz en una expresión alfanumérica contenida en una variable de tipo *String* , que se visualisa luego en una caja de texto. Esta opración se puede realizar con la función siguiente:

```
Public Function FormatoPR(ByRef x0() As String) As String
        ' PRESENTACIÓN DE UN POLINOMIO EN LA PANTALLA
        Dim i As Integer, gx As Integer, ax As String, cx As String
        gx = UBound(x0())
        For i = 0 To gx
                If x0(i) <> "0" Then
                    If i = 0 Then
                        If x0(0) <> "1" And x0(0) <> "-1" Then
                            cx = x0(0)
                        Else
                            If gx <> 0 Then
                                If x0(0) = "-1" Then cx = "-"
                            Else
                                If x0(0) = "-1" Then cx = Str$(-1) Else cx = Mid$(Str$(1), 2)
                            End If
                        End If
                        If gx <> 0 Then
                            If gx = 1 Then
                                cx = cx + " X"
                            Else
                                cx = cx + " X^" + Mid$(Str$(gx), 2)
                            End If
                        End If
                    Else
                        If Left$(x0(i), 1) = "-" Then ax = " - " Else ax = " + "
                        If (x0(i) <> "1" And x0(i) <> "-1") Or i = gx Then
```

```
                    If Left$(x0(i), 1) = "-" Then
                        ax = ax + Mid$(x0(i), 2)
                    Else
                        ax = ax + x0(i)
                    End If
                End If
                If gx > 1 Then
                    If i < gx - 1 Then
                        ax = ax + " X^": ax = ax + Mid$(Str$(gx - i), 2)
                    Else
                        If i = gx - 1 Then ax = ax + " X"
                    End If
                End If
                cx = cx + ax: ax = ""
            End If
        End If
    Next i
    FormatoPR = cx
End Function
```

Cuando los polinomios tienen coeficientes fraccionarios, al multiplicar cada polinomio con el mínimo común múltiplo de los denominadores de los coeficientes, se obtienen dos polinomios con coeficientes enteros y así se puede aplicar en éste caso también el código anterior para hallar un máximo común divisor o un mínimo común múltiplo de los polinomios.

Igualmente que con los polinomios con coeficientes de Z, se pueden operar con los polinomios con coeficientes de $Z[i]$ (enteros de Gauss). A continuación se exponen las funciones necesarias para efectuar estas opraciones en un ordenador (en el Lenguaje Visual-Basic):

```
Public Function SumaPC(ByRef p11() As Double, ByRef p12() As Double, ByRef p21() As Double, ByRef p22() As Double) As Variant
    Dim g1 As Integer, g2 As Integer, gp As Integer, gg As Integer, sw As Integer
    Dim pt1() As Double, pt2() As Double, r() As Double, gr As Integer
    g1 = UBound(p11()): g2 = UBound(p21())
    If g1 >= g2 Then gp = g1 Else gp = g2
    ReDim pt1(gp), pt2(gp)
    gg = Abs(gp - g1)
    For i = 0 To g1
        pt1(i + gg) = p11(i): pt2(i + gg) = p12(i)
    Next i
    gg = Abs(gp - g2)
    For i = 0 To g2
        pt1(i + gg) = pt1(i + gg) + p21(i)
        pt2(i + gg) = pt2(i + gg) + p22(i)
    Next i
    'Grado de la suma
    For i = 0 To gp
        If pt1(i) <> 0 Or pt2(i) <> 0 Then
            sw = 1: Exit For
        End If
    Next i
    If sw = 0 Then i = i - 1
    gr = gp - i
    ReDim r(gr, 2)
    For j = 0 To gr
        r(j, 1) = pt1(i + j): r(j, 2) = pt2(i + j)
    Next j
    SumaPC = r()
End Function
‘ - - -
```

```
        For i = 0 To gp
                If pt1(i) <> 0 Or pt2(i) <> 0 Then
                        sw = 1: Exit For
                End If
        Next i
        If sw = 0 Then i = i - 1
        gr = gp - i
        ReDim r(gr, 2)
        For j = 0 To gr
                r(j, 1) = pt1(i + j): r(j, 2) = pt2(i + j)
        Next j
        RestPC = r()
End Function
' - - -
```

```
                    m2(1) = rr(1, 1): m2(2) = rr(1, 2)
                    For k = i To i + g2
                        rr() = MultNC(c21(k - i), c22(k - i), m2(1), m2(2))
                        rr() = ResNC(c11(k), c12(k), rr(1), rr(2))
                        c11(k) = rr(1): c12(k) = rr(2)
                    Next k
                End If
            Next i
            j0 = g1 - g2 + 1: j = j0
            For i = j0 To g1
                If c11(i) = 0 And c12(i) = 0 Then
                    If s1 = 1 Then
                        z11(i - j) = c11(i): z12(i - j) = c12(i): gz = gz + 1
                    Else
                        j = j + 1
                    End If
                Else
                    z11(i - j) = c11(i): z12(i - j) = c12(i): gz = gz + 1
                    If s1 = 0 Then s1 = s1 + 1
                End If
            Next i
            sw = 0
            For i = 0 To gz - 1
                If z11(i) <> 0 Or z12(i) <> 0 Then
                    ReDim t1(gz - 1), t2(gz - 1)
                    For j = 0 To gz - 1
                        t1(j) = z11(j): t2(j) = z12(j)
                    Next j
                    c11() = c21(): c12() = c22(): c21() = t1(): c22() = t2()
                    g1 = UBound(c11()): g2 = UBound(c21())
                    w() = rutinaC(c21(), c22())
                    For k = 0 To g2
                        c21(k) = w(k, 1): c22(k) = w(k, 2)
                    Next k
                    sw = 1: Exit For
                End If
            Next i
            If sw = 0 Then Exit Do
        Loop
        g2 = UBound(c21())
        w() = rutinaC(c21(), c22())
        McdPZiA = w()
End Function
' - - -
```

```
                                        j = j + 1
                                End If
                        Else
                                z11(i - j) = c11(i): z12(i - j) = c12(i): gz = gz + 1
                                If s1 = 0 Then s1 = s1 + 1
                        End If
                Next i
                sw = 0
                For i = 0 To gz - 1
                        If z11(i) <> 0 Or z12(i) <> 0 Then
                                ReDim t1(gz - 1), t2(gz - 1)
                                For j = 0 To gz - 1
                                        t1(j) = z11(j): t2(j) = z12(j)
                                Next j
                                c11() = c21(): c12() = c22(): c21() = t1(): c22() = t2()
                                g1 = UBound(c11()): g2 = UBound(c21())
                                w() = rutinaC(c21(), c22())
                                For k = 0 To g2
                                        c21(k) = w(k, 1): c22(k) = w(k, 2)
                                Next k
                                sw = 1: Exit For
                        End If
                Next i
                If sw = 0 Then Exit Do
        Loop
        g2 = UBound(c21())
        w() = rutinaC(c21(), c22())
        For k = 0 To g2
                c21(k) = w(k, 1): c22(k) = w(k, 2)
        Next k
        re = FormatoPolinomioEG(c21(), c22())
        McdPZiB = re
End Function
' ----------------------------------------------------------------------------------------------
Public Function rutinaC(ByRef c21() As Double, ByRef c22() As Double) As Variant
        Dim i As Integer, r01 As Double, r02 As Double, g2 As Integer, rr() As Double
        Dim t21() As Double, t22() As Double, res() As Double
        t21() = c21(): t22() = c22()
        g2 = UBound(t21()): r01 = t21(0): r02 = t22(0)
        ReDim res(g2, 2)
        For i = 1 To g2
                If t21(i) <> 0 Or t22(i) <> 0 Then
                        rr() = MCDEG(t21(i), t22(i), r01, r02)
                        r01 = rr(1): r02 = rr(2)
                        If r01 = 1 And r02 = 0 Then Exit For
                End If
        Next i
        If r01 <> 1 Or r02 <> 0 Then
                For i = 0 To g2
                        rr() = DivNC(t21(i), t22(i), r01, r02)
                        res(i, 1) = rr(1): res(i, 2) = rr(2)
                Next i
        Else
                For i = 0 To g2
                        res(i, 1) = c21(i): res(i, 2) = c22(i)
                Next i
        End If
        rutinaC = res()
End Function
' ----------------------------------------------------------------------------------------------
Public Function McmPZiA(ByRef p11() As Double, ByRef p12() As Double, ByRef p21() As Double, ByRef p22() As Double) As Variant
        Dim i As Integer, gpr As Integer, gmcd As Integer, rr() As Double
        Dim pr1() As Double, pr2() As Double, mcd1() As Double, mcd2() As Double
        rr() = ProdPC(p11(), p12(), p21(), p22())   ' Producto de los polinomios
        gpr = UBound(rr(), 1)
        ReDim pr1(gpr), pr2(gpr)
        For i = 0 To gpr
                pr1(i) = rr(i, 1): pr2(i) = rr(i, 2)
        Next i
        rr() = McdPZiA(p11(), p12(), p21(), p22())   ' MCD de los polinomios
        gmcd = UBound(rr(), 1)
        ReDim mcd1(gmcd), mcd2(gmcd)
        For i = 0 To gmcd
                mcd1(i) = rr(i, 1): mcd2(i) = rr(i, 2)
        Next i
        ' División del producto por el MCD.
        rr() = DivExPEG(pr1(), pr2(), mcd1(), mcd2())
         McmPZiA = rr()
End Function
' ----------------------------------------------------------------------------------------------
Public Function McmPZiB(ByRef p11() As Double, ByRef p12() As Double, ByRef p21() As Double, ByRef p22() As Double) As String
```

```
        Dim i As Integer, gpr As Integer, gmcd As Integer, gmcm As Integer
        Dim u11() As Double, u12() As Double, rr() As Double, re As String
        Dim pr1() As Double, pr2() As Double, mcd1() As Double, mcd2() As Double
        rr() = ProdPC(p11(), p12(), p21(), p22()) ' Producto de los polinomios
        gpr = UBound(rr(), 1)
        ReDim pr1(gpr), pr2(gpr)
        For i = 0 To gpr
                pr1(i) = rr(i, 1): pr2(i) = rr(i, 2)
        Next i
        rr() = McdPZiA(p11(), p12(), p21(), p22())  ' MCD de los polinomios
        gmcd = UBound(rr(), 1)
        ReDim mcd1(gmcd), mcd2(gmcd)
        For i = 0 To gmcd
                mcd1(i) = rr(i, 1): mcd2(i) = rr(i, 2)
        Next i
         ' División del producto por el MCD.
        rr() = DivExPEG(pr1(), pr2(), mcd1(), mcd2())
        gmcm = UBound(rr(), 1)
        ReDim u11(gmcm), u12(gmcm)
        For i = 0 To gmcm
                u11(i) = rr(i, 1): u12(i) = rr(i, 2)
        Next i
        re = FormatoPolinomioEG(u11(), u12())
        McmPZiB = re
End Function
'----------------------------------------------------------------------------------------------------
Public Function DivExPEG(ByRef p11() As Double, ByRef p12() As Double, ByRef p21() As Double, ByRef p22() As Double) As Variant
        ' ¡El primer polinomio tiene que ser divisible con el segundo en el conjunto de los polinomios con coeficientes de Z[i]!
        Dim i As Integer, k As Integer, gm As Integer, gpr As Integer, gmcd As Integer
        Dim m21 As Double, m22 As Double, u11() As Double, u12() As Double, rr() As Double
        Dim mp1 As Double, mp2 As Double, m11() As Double, m12() As Double, res() As Double
        Dim pr1() As Double, pr2() As Double, gp1 As Integer, gp2 As Integer
        gp1 = UBound(p11()): gp2 = UBound(p21())
        pr1() = p11(): pr2() = p12()
        gm = gp1 - gp2
        ReDim u11(gm), u12(gm), m11(gm + 1), m12(gm + 1)
        m11(gm + 1) = 1: m12(gm + 1) = 0
        For i = 0 To gm
                If pr1(i) <> 0 Or pr2(i) <> 0 Then
                        rr() = MCMEG(pr1(i), pr2(i), p21(0), p22(0))
                        mp1 = rr(1): mp2 = rr(2)
                        rr() = DivEEG(mp1, mp2, pr1(i), pr2(i))
                        m11(i) = rr(1, 1): m12(i) = rr(1, 2)
                        If m11(i) <> 1 Or m12(i) <> 0 Then
                                For k = i To gp1
                                        rr() = MultNC(pr1(k), pr2(k), m11(i), m12(i))
                                        pr1(k) = rr(1): pr2(2) = rr(2)
                                Next k
                        End If
                        rr() = DivEEG(pr1(i), pr2(i), p21(0), p22(0))
                        u11(i) = rr(1, 1): u12(i) = rr(1, 2)
                        For k = i To i + gp2
                                rr() = MultNC(p21(k - i), p22(k - i), u11(i), u12(i))
                                rr() = ResNC(pr1(k), pr2(k), rr(1), rr(2))
                                pr1(k) = rr(1): pr2(k) = rr(2)
                        Next k
                Else
                        m11(i) = 1: m12(i) = 0: u11(i) = 0: u12(i) = 0
                End If
        Next i
        For k = 0 To gm
                For i = 0 To k
                        rr() = MultNC(u11(i), u12(i), m11(k + 1), m12(k + 1))
                        u11(i) = rr(1): u12(i) = rr(2)
                Next i
        Next k
        ReDim res(gm, 2)
        For i = 0 To gm
                res(i, 1) = u11(i): res(i, 2) = u12(i)
        Next i
        DivExPEG = res()
End Function
'----------------------------------------------------------------------------------------------------
Public Function FormatoPolinomioEG(ByRef z1() As Double, z2() As Double) As String
        'Formato de polinomios coen coeficientes enteros de Gauss (tipo Double)
        Dim i As Integer, j As Integer, gx As Integer, cd As String, cm As String
        gx = UBound(z1())
        For i = 0 To gx
                If z1(i) <> 0 Or z2(i) <> 0 Then
                        If i = 0 Then
                                If z2(0) = 0 Then
```

```
                If Abs(z1(0)) <> 1 Then
                        cm = Str$(z1(0))
                Else
                        If gx <> 0 Then
                                If z1(0) = -1 Then
                                        cm = "-"
                                End If
                        Else
                                If z1(0) = -1 Then
                                        cm = Str$(-1)
                                Else
                                        cm = Mid$(Str$(1), 2)
                                End If
                        End If
                End If
        Else
                If gx <> 0 Then
                        If z1(0) <> 0 Then
                                cm = cm + "(" + Str$(z1(0))
                        End If
                        If Abs(z2(0)) <> 1 Then
                                If z1(0) <> 0 Then
                                        If z2(0) > 0 Then
                                                If z1(0) <> 0 Then
                                                        cm = cm + " + " + Mid$(Str$(z2(0)), 2) + " i )"
                                                Else
                                                        cm = cm + Mid$(Str$(z2(0)), 2) + " i )"
                                                End If
                                        Else
                                                cm = cm + " - " + Mid$(Str$(z2(0)), 2) + " i )"
                                        End If
                                Else
                                        cm = cm + Str$(z2(0)) + " i"
                                End If
                        Else
                                If z2(0) = 1 Then
                                        If z1(0) <> 0 Then
                                                cm = cm + " + i )"
                                        Else
                                                cm = cm + " i "
                                        End If
                                Else
                                        If z1(0) <> 0 Then
                                                cm = cm + " - i )"
                                        Else
                                                cm = cm + " - i "
                                        End If
                                End If
                        End If
                Else
                        If z1(0) <> 0 Then cm = cm + Str$(z1(0))
                        If Abs(z2(0)) <> 1 Then
                                If z2(0) > 0 Then
                                        If z1(0) <> 0 Then
                                                cm = cm + " + " + Mid$(Str$(z2(0)), 2) + " i"
                                        Else
                                                cm = cm + Mid$(Str$(z2(0)), 2) + " i"
                                        End If
                                Else
                                        cm = cm + " - " + Mid$(Str$(z2(0)), 2) + " i"
                                End If
                        Else
                                If z2(0) = 1 Then
                                        cm = cm + "+ i"
                                Else
                                        cm = cm + "- i"
                                End If
                        End If
                End If
        End If
        If gx <> 0 Then
                If gx = 1 Then
                        cm = cm + " X "
                Else
                        cm = cm + " X^" + Mid$(Str$(gx), 2)
                End If
        End If
Else
        If Abs(z2(i)) <> 0 Then
                If i < gx Then
                        If z1(i) <> 0 Then
```

```
                    cd = cd + " + (" + Str$((z1(i)))
                    If Abs(z2(i)) <> 1 Then
                        If z2(i) > 0 Then
                            cd = cd + " + " + Mid$(Str$(z2(i)), 2) + " i )"
                        Else
                            If z2(i) < 0 Then
                                cd = cd + " - " + Mid$(Str$(z2(i)), 2) + " i )"
                            End If
                        End If
                    Else
                        If z2(i) = 1 Then
                            cd = cd + " + i )"
                        Else
                            cd = cd + " - i )"
                        End If
                    End If
                Else
                    If Abs(z2(i)) <> 1 Then
                        If z2(i) < 0 Then
                            cd = cd + " - " + Mid$(Str$(z2(i)), 2) + " i"
                        Else
                            If z2(i) > 0 Then
                                cd = cd + " + " + Mid$(Str$(z2(i)), 2) + " i"
                            End If
                        End If
                    Else
                        If z2(i) = 1 Then
                            cd = cd + " + i"
                        Else
                            cd = cd + " - i"
                        End If
                    End If
                End If
            Else
                If z1(i) > 0 Then
                    cd = cd + " + " + Mid$(Str$(z1(i)), 2)
                Else
                    If z1(i) < 0 Then
                        cd = cd + " - " + Mid$(Str$(z1(i)), 2)
                    End If
                End If
                If Abs(z2(i)) <> 1 Then
                    If z2(i) > 0 Then
                        cd = cd + " + " + Mid$(Str$(z2(i)), 2) + " i"
                    Else
                        If z2(i) < 0 Then
                            cd = cd + " - " + Mid$(Str$(z2(i)), 2) + " i"
                        End If
                    End If
                Else
                    If z2(i) = 1 Then
                        cd = cd + " + i"
                    Else
                        cd = cd + " - i"
                    End If
                End If
            End If
        Else
            If Abs(z1(i)) <> 0 Then
                If Abs(z1(i)) <> 1 Then
                    If z1(i) < 0 Then
                        cd = cd + " - " + Mid$(Str$(z1(i)), 2)
                    Else
                        cd = cd + " + " + Mid$(Str$(z1(i)), 2)
                    End If
                Else
                    If z1(i) = 1 Then
                        cd = cd + " + "
                    Else
                        cd = cd + " - "
                    End If
                    cd = cd + "1"
                End If
            End If
        End If
        If gx > 1 Then
            If i < gx - 1 Then
                cd = cd + " X^" + Mid$(Str$(gx - i), 2)
            Else
                If i = gx - 1 Then
                    cd = cd + " X "
```

```
                                        End If
                                    End If
                                End If
                                cm = cm + cd: cd = ""
                            End If
                        Else
                            If gx = 0 Then cm = "0"
                        End If
            Next i
        FormatoPolinomioEG = cm
End Function
```

Las funciones McdPZiB y McmPZiB devuelven el resultado de manera editada, mientras que McdPZiA y McmPZiA devuelven los coeficientes del polinomio resultado, pudiendo hacer cálculos con el resultado. La función siguiente efectua una división exacta y divide el cociente con el máximo común divisor de los coeficientes:

```
Public Function DivEspPEG(ByRef p11() As Double, ByRef p12() As Double, ByRef p21() As Double, ByRef p22() As Double) As Variant
        'Ee supone que el cociente exite.
         'La diferencia entre el cociente obtenido y el real difieren en un factor que es un entero de Gauss.
        Dim i As Integer, k As Integer, gm As Integer, gpr As Integer, gmcd As Integer, res() As Double
        Dim m21 As Double, m22 As Double, u11() As Double, u12() As Double, m11() As Double, m12() As Double
        Dim r01  As Double, r02 As Double, rr() As Double, mp1 As Double, mp2 As Double
        Dim pr1() As Double, pr2() As Double, gp1 As Integer, gp2 As Integer, sw As Integer
        gp1 = UBound(p11()): gp2 = UBound(p21())
        pr1() = p11(): pr2() = p12()
        gm = gp1 - gp2
        ReDim u11(gm), u12(gm), m11(gm + 1), m12(gm + 1)
        m11(gm + 1) = 1: m12(gm + 1) = 0
        For i = 0 To gm
                If pr1(i) <> 0 Or pr2(i) <> 0 Then
                        rr() = MCMEG(pr1(i), pr2(i), p21(0), p22(0))
                        mp1 = rr(1): mp2 = rr(2)
                        rr() = DivEEG(mp1, mp2, pr1(i), pr2(i))
                        m11(i) = rr(1, 1): m12(i) = rr(1, 2)
                        If m11(i) <> 1 Or m12(i) <> 0 Then
                                For k = i To gp1
                                        rr() = MultNC(pr1(k), pr2(k), m11(i), m12(i))
                                        pr1(k) = rr(1): pr2(k) = rr(2)
                                Next k
                        End If
                        rr() = DivEEG(pr1(i), pr2(i), p21(0), p22(0))
                        u11(i) = rr(1, 1): u12(i) = rr(1, 2)
                        For k = i To i + gp2
                                rr() = MultNC(p21(k - i), p22(k - i), u11(i), u12(i))
                                rr() = ResNC(pr1(k), pr2(k), rr(1), rr(2))
                                pr1(k) = rr(1): pr2(k) = rr(2)
                        Next k
                Else
                        m11(i) = 1: m12(i) = 0: u11(i) = 0: u12(i) = 0
                End If
        Next i
        For k = 0 To gm
                For i = 0 To k
                        rr() = MultNC(u11(i), u12(i), m11(k + 1), m12(k + 1))
                        u11(i) = rr(1): u12(i) = rr(2)
                Next i
        Next k
        i = 1: r01 = u11(0): r02 = u12(0)
        Do Until i > gm
                If u11(i) <> 0 Or u12(i) <> 0 Then
                        rr() = MCDEG(u11(i), u12(i), r01, r02)
                        r01 = rr(1): r02 = rr(2)
                        If Abs(r01) = 1 And r02 = 0 Then
                                sw = 1: Exit Do
                        End If
                End If
                i = i + 1
        Loop
        ReDim res(gm, 2)
        If sw = 0 Then
                For i = 0 To gm
                        rr() = DivEEG(u11(i), u12(i), r01, r02)
                        res(i, 1) = rr(1, 1): res(i, 2) = rr(1, 2)
                Next i
        Else
                For i = 0 To gm
                        res(i, 1) = u11(i): res(i, 2) = u12(i)
                Next i
```

```
        End If
        DivEspPEG = res()
End Function
```

Finalmente, para introducir en un ordenador el polinomio

$$P(X)=(4-3i)X^4+(-1+5i)X^3+(-7-3i)X^2+6iX-3+5i$$

, se recomienda de proceder de la manera siguiente: En una caja de textos hay que introducir la expresión siguiente: 4,–3,–1,5,–8,–7,–3,0,6,–3,5 (sin espacios).

Ejemplo 5.3: Dados los polinomios

$$P(X)=(2+4i)X^3+(1+5i)X^2+(-54-42i)X+130+26i$$

$$Q(X)=2X^3+(10-2i)X^2+(-12-5i)X-65-13i$$

, las funciones McdPZiB y McmPZiB devuelven los resultados siguientes:

$$R(X)=mcd\,[P(X),Q(X)]=(1+i)X^2+(4+i)X-13-13i$$

$$S(X)=mcm\,[P(X),Q(X)]=(6+2i)X^4+(20+12i)X^3+(-83+25i)X^2+(-90-122i)X+442-182i$$

Utilizando la función DivExPEG se pueden determinar los cocientes siguientes:

$$P(X)/R(X)=(3+i)X-6+4i\,.$$

$$Q(X)/R(X)=(1-i)X+3-2i$$

Ejemplo 5.4: Cosiderando los polinomios

$$P(X)=[(2-3i)X+1-7i](3iX^2-1)=(9+6i)X^3+(21+3i)X^2+(-2+3i)X-1+7i$$

$$Q(X)=[(2-3i)X+1-7i]\cdot\left[X^3-(3+7i)X+1-i\right]=(2-3i)X^4+(1-7i)X^3+(-27-5i)X^2+(-53+9i)X-6-8i$$

las funciones anteriores devuelven el resultado siguiente:

$$mcd\,[P(X),Q(X)]=(-3-2i)X-7-i$$

$$mcm\,[P(X),Q(X)]=(-6+9i)X^6+(-3+21i)X^5+(78+13i)X^4+(152-28i)X^3+(13+51i)X^2+(9+53i)X-8+6i$$

Ejemplo 5.5: Multiplicando los polinomios

$$a(X)=(2+3i)X-5+12i \quad y \quad b(X)=(10+5i)X-17+23i$$

, por el polinomio

$$c(X)=(2137-1735i)X^2+(104-830i)X+2456+534i$$

, se obtienen los polinomios

$$P(X)=(9479+2941i)X^3+(12833+32971i)X^2+(12750+13834i)X-18688+26802i$$

$$Q(X)=(30045-6665i)X^3+(8766+70866i)X^2+(39212+34122i)X-54034+47410i$$

Naturalmente se espera que el máximo común divisor de los polinomios $P(X)$ y $Q(X)$ sea el polinomio $c(X)$. Calculando un máximo común divisor de los polinomios $P(X)$ y $Q(X)$, utilizando las funciones McdPZiA ó McdPZiB se obtene el resultado:

$$c'(X)=(1936+201i)X^2+(467-363i)X+961+1495i\,.$$

No se trata de un error puesto que efectuando las divisiones euclideas entre los coeficientes de los términos de mismo grado (utilizando la función DivEEGG) se obtiene que:

$$2137-1735i=(1936+201i)\cdot(1-i),\ 104-830i=(467-363i)\cdot(1-i),\ 2456++534i=(961+1495i)\cdot(1-i)$$

, y por tanto $c(X)=c'(X)\cdot(1-i)$, es decir tanto $c(X)$ como $c'(X)$ son máximo cómun divisor de $P(X)$ y $Q(X)$. Las funciones McmPZiA ó McmPZiB devuelven el resultado siguiente:

$$mcm(P(X),Q(X))=z_0X^4+z_1X^3+z_2X^2+z_3X+z_4$$

, donde

$z_0=156890-3280i$, $z_1=296584+827206i$, $z_2=-981422+854906i$, $z_3=-623170+1088474i$, $z_4=-1184208+586708$.

Para poder calcular un máximo común divisor o un mínimo común múltiplo de dos polinomios con coeficientes pertenecientes a $Z[i]$, cuyas partes reales e imaginarias son enteros largos, se exponen las funciones siguientes:

```
Public Function SumaPCG(ByRef p11() As String, ByRef p12() As String, ByRef p21() As String, ByRef p22() As String) As Variant
        Dim g1 As Integer, g2 As Integer, gp As Integer, gg As Integer, sw As Integer, x(2) As String
        Dim pt1() As String, pt2() As String, r() As String, gr As Integer, i As Integer
```

```
        g1 = UBound(p11()): g2 = UBound(p21())
        If g1 >= g2 Then gp = g1 Else gp = g2
        ReDim pt1(gp), pt2(gp)
        gg = Abs(gp - g1)
        For i = 0 To g1
                pt1(i + gg) = p11(i): pt2(i + gg) = p12(i)
        Next i
        gg = Abs(gp - g2)
        For i = 0 To g2
                x(1) = pt1(i + gg): x(2) = p21(i): pt1(i + gg) = Sumar(x(), 7)
                x(1) = pt2(i + gg): x(2) = p22(i): pt2(i + gg) = Sumar(x(), 7)
        Next i
        Grado de la suma
        For i = 0 To gp
                If pt1(i) <> "0" Or pt2(i) <> "0" Then
                        sw = 1: Exit For
                End If
        Next i
        If sw = 0 Then i = i - 1
        gr = gp - i
        ReDim r(gr, 2)
        For j = 0 To gr
                r(j, 1) = pt1(i + j): r(j, 2) = pt2(i + j)
        Next j
        SumaPCG = r()
End Function
' - - -
```

```
        x(1) = u2: x(2) = v1: p2 = Multiplicar(x(), 7)
        x(1) = p1: x(2) = p2: pc(2) = Sumar(x(), 7)
        MultNCG = pc()
End Function
'-----------------------------------------------------------------------------------------------
Public Function FixNG(ByVal u As String) As String
        Dim v As String, pp As String, k As Integer, x(2) As String
        If u = "-1" Then FixNG = "-1": Exit Function
        If u = "1" Then FixNG = "1": Exit Function
        If u = "0" Then FixNG = "0": Exit Function
        If Left$(u, 2) = "0." Then FixNG = "0": Exit Function
        If Left$(u, 3) = "-0." Then FixNG = "0": Exit Function
        For k = 1 To Len(u)
                pp = Right$(Left$(u, k), 1)
                If pp = "." Then
                        v = Left$(u, k - 1)
                        If Left$(u, 1) <> "-" Then
                                FixNG = v: Exit Function
                        Else
                                x(1) = v: x(2) = "1": FixNG = Sumar(x(), 7)
                                Exit Function
                        End If
                End If
        Next k
        FixNG = u
End Function
'-----------------------------------------------------------------------------------------------
Public Function McdPZiAG(ByRef p11() As String, ByRef p12() As String, ByRef p21() As String, ByRef p22() As String) As Variant
        Dim i As Integer, j As Integer, k As Integer, s1 As Integer, j0 As Integer, r01 As String, r02 As String
        Dim g1 As Integer, g2 As Integer, gz As Integer, sw As Integer, re As String, z() As String
        Dim c11() As String, c12() As String, c21() As String, c22() As String, rr() As String
        Dim mp() As String, m11(2) As String, m2(2) As String, t1() As String, t2() As String
        Dim m12() As String, z11() As String, z12() As String, w() As String
        c11() = p11(): c12() = p12(): c21() = p21(): c22() = p22()
        g1 = UBound(p11()): g2 = UBound(p21())
        ReDim z11(g2), z12(g2)
        Do
                gz = 0: s1 = 0
                For i = 0 To g1 - g2
                        If c11(i) <> "0" Or c12(i) <> "0" Then
                                mp() = MCMEGG(c11(i), c12(i), c21(0), c22(0))
                                rr() = DivEEGG(mp(1), mp(2), c11(i), c12(i))
                                m11(1) = rr(1, 1): m11(2) = rr(1, 2)
                                If m11(1) <> "1" Or m11(2) = "0" Then
                                        For k = i To g1
                                                rr() = MultNCG(c11(k), c12(k), m11(1), m11(2))
                                                c11(k) = rr(1): c12(k) = rr(2)
                                        Next k
                                End If
                                rr() = DivEEGG(c11(i), c12(i), c21(0), c22(0))
                                m2(1) = rr(1, 1): m2(2) = rr(1, 2)
                                For k = i To i + g2
                                        rr() = MultNCG(c21(k - i), c22(k - i), m2(1), m2(2))
                                        rr() = ResNCG(c11(k), c12(k), rr(1), rr(2))
                                        c11(k) = rr(1): c12(k) = rr(2)
                                Next k
                        End If
                Next i
                j0 = g1 - g2 + 1: j = j0
                For i = j0 To g1
                        If c11(i) = "0" And c12(i) = "0" Then
                                If s1 = 1 Then
                                        z11(i - j) = c11(i): z12(i - j) = c12(i): gz = gz + 1
                                Else
                                        j = j + 1
                                End If
                        Else
                                z11(i - j) = c11(i): z12(i - j) = c12(i): gz = gz + 1
                                If s1 = 0 Then s1 = s1 + 1
                        End If
                Next i
                sw = 0
                For i = 0 To gz - 1
                        If z11(i) <> "0" Or z12(i) <> "0" Then
                                ReDim t1(gz - 1), t2(gz - 1)
                                For j = 0 To gz - 1
                                        t1(j) = z11(j): t2(j) = z12(j)
                                Next j
                                c11() = c21(): c12() = c22(): c21() = t1(): c22() = t2()
                                g1 = UBound(c11()): g2 = UBound(c21())
                                w() = rutinaCG(c21(), c22())
```

```
                        For k = 0 To g2
                                c21(k) = w(k, 1): c22(k) = w(k, 2)
                        Next k
                        sw = 1: Exit For
                End If
        Next i
        If sw = 0 Then Exit Do
    Loop
    g2 = UBound(c21())
    w() = rutinaCG(c21(), c22())
  McdPZiAG = w()
End Function
' - - -
```

```
        McdPZiBG = re
End Function
' - - -
```

```
        ReDim pr1(gp1), pr2(gp1)
        For i = 0 To gp1
                pr1(i) = p11(i): pr2(i) = p12(i)
        Next i
        gm = gp1 - gp2
        ReDim u11(gm), u12(gm), m11(gm + 1), m12(gm + 1)
        m11(gm + 1) = "1": m12(gm + 1) = "0"
        For i = 0 To gm
                If pr1(i) <> "0" Or pr2(i) <> "0" Then
                        rr() = MCMEGG(pr1(i), pr2(i), p21(0), p22(0))
                        mp1 = rr(1): mp2 = rr(2)
                        rr() = DivEEGG(mp1, mp2, pr1(i), pr2(i))
                        m11(i) = rr(1, 1): m12(i) = rr(1, 2)
                        If m11(i) <> "1" Or m12(i) <> "0" Then
                                For k = i To gp1
                                        rr() = MultNCG(pr1(k), pr2(k), m11(i), m12(i))
                                        pr1(k) = rr(1): pr2(k) = rr(2)
                                Next k
                        End If
                        rr() = DivEEGG(pr1(i), pr2(i), p21(0), p22(0))
                        u11(i) = rr(1, 1): u12(i) = rr(1, 2)
                        For k = i To i + gp2
                                rr() = MultNCG(p21(k - i), p22(k - i), u11(i), u12(i))
                                rr() = ResNCG(pr1(k), pr2(k), rr(1), rr(2))
                                pr1(k) = rr(1): pr2(k) = rr(2)
                        Next k
                Else
                        m11(i) = "1": m12(i) = "0": u11(i) = "0": u12(i) = "0"
                End If
        Next i
        For k = 0 To gm
                For i = 0 To k
                        rr() = MultNCG(u11(i), u12(i), m11(k + 1), m12(k + 1))
                        u11(i) = rr(1): u12(i) = rr(2)
                Next i
        Next k
        ReDim res(gm, 2)
        For i = 0 To gm
                res(i, 1) = u11(i): res(i, 2) = u12(i)
        Next i
        DivExPCEGG = res()
End Function
' - - -
```

```
		For i = 0 To k
			rr() = MultNCG(u11(i), u12(i), m11(k + 1), m12(k + 1))
			u11(i) = rr(1): u12(i) = rr(2)
		Next i
	Next k
	i = 1: r01 = u11(0): r02 = u12(0)
	Do Until i > gm
		If u11(i) <> "0" Or u12(i) <> "0" Then
			rr() = MCDEGG(u11(i), u12(i), r01, r02)
			r01 = rr(1): r02 = rr(2)
			If (r01 = "1" Or r01 = "-1") And r02 = "0" Then
				sw = 1: Exit Do
			End If
		End If
		i = i + 1
	Loop
	ReDim res(gm, 2)
	If sw = 0 Then
		For i = 0 To gm
			rr() = DivEEGG(u11(i), u12(i), r01, r02)
			res(i, 1) = rr(1, 1): res(i, 2) = rr(1, 2)
		Next i
	Else
		For i = 0 To gm
			res(i, 1) = u11(i): res(i, 2) = u12(i)
		Next i
	End If
	DivEspPCEGG = res()
End Function
' - - -
```

```
                                        cm = cm + " - i "
                                End If
                        End If
                End If
        Else
                If z1(0) <> "0" Then cm = cm + z1(0)
                If Left$(z2(0), 1) = "-" Then y = Mid$(z2(0), 2) Else y = z2(0)
                If y <> "1" Then
                        If z1(0) <> "0" Then
                                If Left$(z2(0), 1) <> "-" Then
                                        cm = cm + " + " + z2(0) + " i"
                                Else
                                        cm = cm + " - " + Mid$(z2(0), 2) + " i"
                                End If
                        Else
                                If Left$(z2(0), 1) = "-" Then
                                        cm = cm + " - " + Mid$(z2(0), 2) + " i"
                                Else
                                        cm = cm + z2(0) + " i"
                                End If
                        End If
                Else
                        If z2(0) = "1" Then
                                cm = cm + "+i"
                        Else
                                cm = cm + "- i"
                        End If
                End If
        End If
End If
If gx <> 0 Then
        If gx = 1 Then
                cm = cm + " X "
        Else
                cm = cm + " X^" + Mid$(Str$(gx), 2)
        End If
End If
Else
        If Left$(z2(i), 1) = "-" Then y = Mid$(z2(i), 2) Else y = z2(i)
        If y <> "0" Then
                If i < gx Then
                        If z1(i) <> "0" Then
                                cd = cd + " + (" + z1(i)
                                If y <> "1" Then
                                        If Left$(z2(i), 1) <> "-" And z2(i) <> "0" Then
                                                cd = cd + " + " + z2(i) + " i) "
                                        Else
                                                If Left$(z2(i), 1) = "-" Then
                                                        cd = cd + " - " + Mid$(z2(i), 2) + " i )"
                                                End If
                                        End If
                                Else
                                        If z2(i) = "1" Then
                                                cd = cd + " + i )"
                                        Else
                                                cd = cd + " - i )"
                                        End If
                                End If
                        Else
                                If Left$(z2(i), 1) = "-" Then y = Mid$(z2(i), 2) Else y = z2(i)
                                If y <> "1" Then
                                        If Left$(z2(i), 1) = "-" Then
                                                cd = cd + " - " + Mid$(z2(i), 2) + " i"
                                        Else
                                                If Left$(z2(i), 1) <> "-" And z2(i) <> "0" Then
                                                        cd = cd + " + " + z2(i) + " i"
                                                End If
                                        End If
                                Else
                                        If z2(i) = 1 Then
                                                cd = cd + " + i"
                                        Else
                                                cd = cd + " - i"
                                        End If
                                End If
                        End If
                Else
                        If Left$(z1(i), 1) <> "-" And z1(i) <> "0" Then
                                cd = cd + " + " + z1(i)
                        Else
                                If Left$(z1(i), 1) = "-" Then
```

```
                                        cd = cd + " - " + Mid$(z1(i), 2)
                                    End If
                                End If
                                If Left$(z2(i), 1) = "-" Then y = Mid$(z2(i), 2) Else y = z2(i)
                                If y <> "1" Then
                                    If Left$(z2(i), 1) <> "-" And z2(i) <> "0" Then
                                        cd = cd + " + " + z2(i) + " i"
                                    Else
                                        If Left$(z2(i), 1) = "-" Then
                                            cd = cd + " - " + Mid$(z2(i), 2) + " i"
                                        End If
                                    End If
                                Else
                                    If z2(i) = "1" Then
                                        cd = cd + " + i"
                                    Else
                                        cd = cd + " - i"
                                    End If
                                End If
                            End If
                        Else
                            If z1(i) <> "0" Then
                                If Left$(z1(i), 1) = "-" Then y = Mid$(z1(i), 2) Else y = z1(i)
                                If y <> "1" Then
                                    If Left$(z1(i), 1) = "-" Then
                                        cd = cd + " - " + Mid$(z1(i), 2)
                                    Else
                                        If z1(i) <> "0" Then cd = cd + " + " + z1(i)
                                    End If
                                Else
                                    If z1(i) = "1" Then
                                        cd = cd + " + "
                                    Else
                                        cd = cd + " - "
                                    End If
                                    cd = cd + "1"
                                End If
                            End If
                        End If
                        If gx > 1 Then
                            If i < gx - 1 Then
                                cd = cd + " X^" + Mid$(Str$(gx - i), 2)
                            Else
                                If i = gx - 1 Then
                                    cd = cd + " X "
                                End If
                            End If
                        End If
                        cm = cm + cd: cd = ""
                    End If
                Else
                    If gx = 0 Then cm = "0"
                End If
            Next i
            FormatoPolinomioEGG = cm
End Function
```

Finalmente, para introducir en un ordenador un polinomio cuyos coeficientes son enteros de Gauss, se recomienda proceder de la manera siguiente. Si por ejemplo

$$C_1(X) = (3-4i)X^4 + (-8-5i)X^3 + 6X - 9 + 2i = C_{11}(x) + iC_{12}(X)$$

, donde

$$C_{12}(X) = 3X^4 - 8X^3 + 6X - 9 \quad \text{y} \quad C_{12}(X) = -4X^4 - 5X^3 + 2$$

, entonces en una caja de texto (por ejemplo Text1) hay que introducir la expresión alfanumeérica $\alpha = 3,-4,-8,-5,0,0,6,0-9,2$ y de aquí separar los coeficientes de los polinomios $C_{11}(x)\, y\, C_{12}(X)$. En el caso de los coeficientes de tipo Double, el código necesario para esto es la siguiente:

```
Private Sub Command1_Click()
        Dim i As Integer, g1 As Integer, t As String
        Dim rr() As Double, caracter As String, nc As Integer
         t = Text1
        For i = 1 To Len(t)
                caracter = Right$(Left$(t, i), 1)
                If caracter = "," Then nc = nc + 1
        Next i
        nc = nc Mod 2
        If nc <> 1 Then
```

```
        MsgBox "Tiene que haber un numero par de coeficientes", 48
        Form1.Text1.SetFocus: Exit Sub
        End If
        rr() = RutinaCoeficientesEG(t)
        g1 = UBound(rr(), 1)
        ReDim c11(g1), c12(g1)
        For i = 0 To g1
                c11(i) = rr(i, 1): c12(i) = rr(i, 2)
        Next i
        If c11(0) = 0 And c12(0) = 0 Then
                MsgBox "La lista de los coeficientes no puede empezar por 0,0,...", 48
                Form1.Text1.SetFocus: Exit Sub
        End If
End Sub
'- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Public Function RutinaCoeficientesZi(ByVal t As String)
        Dim i As Integer, j As Integer, k As Integer, i0 As Integer, g1 As Integer
        Dim nco As Integer, gp As Integer, lt As Integer, bb As String
        Dim pr() As Double, px() As Double
        Dim g0 As Integer, pa As Integer
         '---- Número de las comas en la cadena t
        If Right$(t, 1) <> "," Then t = t + ","
        k = 1: lt = Len(t): nco = 0
        Do
                bb = Right$(Left$(t, k), 1)
                If bb = "," Then: nco = nco + 1
                k = k + 1
                If k > lt Then Exit Do
        Loop
        g0 = nco - 1
         '--- Separación de los coeficientes
        ReDim px(g0)
        k = 1: i = 1: i0 = 0:  j = 0
        Do
                 bb = Right$(Left$(t, k), 1)
                If bb = "," Then
                        j = j + 1
                        px(i0) = Val(Left$(t, k - 1))
                        i = i + 1: i0 = i0 + 1
                        t = Mid$(t, k + 1)
                        k = 1
                Else
                        k = k + 1
                End If
                If j = nco Then Exit Do
        Loop
        ' Polinomio con coeficientes de enteros de Gauss
        gp = (g0 - 1) / 2
        ReDim pr(gp, 2)
        i = 0: k = 0
        For j = 0 To g0
                pa = j Mod 2
                If pa = 0 Then
                        pr(i, 1) = px(j): i = i + 1
                Else
                        pr(k, 2) = px(j): k = k + 1
                End If
        Next j
        RutinaCoeficientesZi = pr()
End Function
```

, donde $c11()$ y $c12()$ son dos variables globales de tipo *Double* , declaradas en *Option Explicit* .

Cuando los coeficientes largos de los polinomios se almacenan en variables de tipo String la entrada de los coeficientes se podría realizar de la manera siguiente:

```
Private Sub Command1_Click()
        Dim i As Integer, rr() As String, g1 As Integer
        Dim caracter As String, nc As Integer
         t = Text1
        For i = 1 To Len(t)
                caracter = Right$(Left$(t, i), 1)
                If caracter = "," Then nc = nc + 1
        Next i
        nc = nc Mod 2
        If nc <> 1 Then
                MsgBox "Tiene que haber un numero par de coeficientes", 48
                Form1.Text1.SetFocus: Exit Sub
        End If
```

```
        rr() = RutinaCoeficientes(t)
        g1 = UBound(rr(), 1)
        ReDim p11(g1), p12(g1)
        For i = 0 To g1
                p11(i) = rr(i, 1)
                p12(i) = rr(i, 2)
        Next i
        If p11(0) = "0" And p12(0) = "0" Then
                MsgBox "La lista de los coeficientes no puede empezar por 0,0,...", 48
                Form1.Text1.SetFocus: Exit Sub
        End If
End Sub
‘ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Public Function RutinaCoeficientesZiG(ByVal t As String)
        Dim i As Integer, j As Integer, k As Integer, i0 As Integer, g0 As Integer
        Dim nco As Integer, gp As Integer, lt As Integer, bb As String
        Dim px() As String, pr() As String, pa As Integer
         '---- Número de las comas en la cadena t
        If Right$(t, 1) <> "," Then t = t + ","
         k = 1: lt = Len(t): nco = 0
        Do
                bb = Right$(Left$(t, k), 1)
                If bb = "," Then nco = nco + 1
                k = k + 1
                If k > lt Then Exit Do
        Loop
        g0 = nco - 1
         '--- Separación de los coeficientes
        ReDim px(g0)
        k = 1: i = 1: i0 = 0:  j = 0
        Do
                bb = Right$(Left$(t, k), 1)
                If bb = "," Then
                        j = j + 1
                        px(i0) = Left$(t, k - 1)
                        i = i + 1: i0 = i0 + 1
                        t = Mid$(t, k + 1)
                        k = 1
                Else
                        k = k + 1
                End If
                If j = nco Then Exit Do
        Loop
        ' Polinomio con coeficientes de enteros de Gauss
        gp = (g0 - 1) / 2
        ReDim pr(gp, 2)
        i = 0: k = 0
        For j = 0 To g0
                pa = j Mod 2
                If pa = 0 Then
                        pr(i, 1) = px(j): i = i + 1
                Else
                        pr(k, 2) = px(j): k = k + 1
                End If
        Next j
        RutinaCoeficientesZiG = pr()
End Function
```

§.6. Cálculo de los ceros reales de los polinomios con coeficentes de $Z[i]$ ó $Z[\varepsilon]$.

Cosiderando polinomios cuyos coeficientes son enteros de Gauss, sea

$$P(X) = (a_0 + b_0 i)X^n + (a_1 + b_1 i)X^{n-1} + \cdots + (a_{n-1} + b_{n-1} i)X + a_n + b_n i \qquad a_i, b_i \in Z \tag{6.1}$$

Entonces

$$P(X) = P_1(X) + iP_2(X) \tag{6.2}$$

, donde

$$P_1(X) = a_0 X^n + a_1 X^{n-1} + \cdots + a_{n-1} X + a_n$$
$$P_2(X) = b_0 X^n + b_1 X^{n-1} + \cdots + b_{n-1} X + b_n$$

, son polinomios reales con coeficientes enteros.

Si α es un cero real del polinomio $P(X)$ y $P_1(X)$ y $P_2(X)$ no son de grado cero, entonces

$$P(\alpha) = 0 \quad \Leftrightarrow \quad P_1(\alpha) + iP_2(\alpha) = 0 \quad \Leftrightarrow \quad P_1(\alpha) = 0 \;\; y \;\; P_2(\alpha) = 0$$

,es decir $P(X)$ y el máximo común divisor $M(X)$ de los polinomios $P_1(X)$ y $P_2(X)$, tienen los mismos ceros reales. Los ceros complejos de $M(X)$ son también ceros del polinomio $P(X)$. El número de los ceros reales distintos del polinomio $P(X)$, se determina según el protocolo es la siguiente: Se expresa $P(X)$ en la forma (5.14) y se calcula $M(X)$. Puesto que $M(X)$ puede tener ceros múltiples hay que calcular la derivada $M'(X)$ y sea $M_1(X)$ el máximo común divisor de $M(X)$ y $M'(X)$. Si $Q(X)=M(X)/M_1(X)$ entonces $Q(X)$ tiene los mismos ceros reales que $P(X)$, pero son simples. Construyendo la serie de polinomios Sturm para el polinomio $Q(X)$ (ver [4]), el número de los ceros reales distintos de $P(X)$ será igual con el número de los ceros simples de $Q(X)$.

Los ceros reales de $P(X)$ se calculan hallando los ceros de $Q(X)$ por el método de Lobachevski. Si se obtienen también ceros complejos para $Q(X)$, estos serán ceros también de $P(X)$.

Ejemplo 6.1: Para hallar el número de las raíces reales del polinomio

$$P(X)=(1+i)X^5+(-8-3i)X^4+(19-6i)X^3+(-9+11i)X^2+15iX-27+18i$$

, se procede de la manera siguiente:

$$P(X)=X^5-8X^4+19X^3-9X^2-27+i\left(X^5-3X^4-6X^3+11X^2+15X+18\right)$$

Por tanto

$$P_1(X)=X^5-8X^4+19X^3-9X^2-27 \quad \text{y} \quad P_2(X)=X^5-3X^4-6X^3+11X^2+15X+18.$$

Utilizando la función CalculoMCDPOL, resulta que

$$M(X)=X^4-5X^3+4X^2+3X+9\ ,\ M'(X)=4X^3-15X^2+8X+3$$

$$M_1(X)=X-3,\ Q(X)=X^3-2X^2-2X-3$$

No es difícil averiguar que los ceros reales del polinomio $Q(X)$ se encuentran en el intervalo $(-4,4)$ y que la serie de polinomios Sturm asociado al polinomio es la siguiente:

$$Q_1(X)=X^3-2X^2-2X+3,\ Q_2(X)=3X^2-4X-2,\ Q_3(X)=20X+31 \text{ y } Q_4(X)=-1.$$

Las sucesiones de los signos de los valores de los polinomios Sturm en los puntos -4 y 4, respectivamente, son:

$-+--$ y $++++-$. Esto indica que tanto el polinomio $Q(X)$ como el polinomio $P(X)$ tienen un cero real. Puesto que en este caso $M(3)=M'(3)=0$ y $M''(3)=-4\neq 0$, 3 es un cero doble para $M(X)$ y simple para $M'(X)$ y $Q(X)$. Así el único cero real del polinomio $P(X)$ es el número 3 y tiene el orden de multiplicidad 2. Luego, puesto que $P(X)$ es divisible con $(X-3)^2$ con el esquema de Ruffini se puede averiguar que:

$$P(X)=(X-3)^2\left[(1+i)X^3+(-2+3i)X^2+(-2+3i)X-3+2i\right].$$

Para hallar los ceros complejos de $P(X)$ se puede intentar factorizar la expresión entre corchetes:

Si se introduce la notación $a=-2+3i$, resulta que $-3+2i=a-(1+i)$ y así,

$$(1+i)X^3+(-2+3i)X^2+(-2+3i)X-3+2i=(1+i)(X^3-1)+a\left(X^2+X+1\right)=$$
$$=\left(X^2+X+1\right)\left[(1+i)(X-1))+a\right]=\left(X^2+X+1\right)\left[(1+i)X-3+2i\right]$$

Por consiguiente, los ceros complejos de $P(X)$ son soluciones de las ecuaciones

$$X^2+X+1=0 \quad \text{o} \quad (1+i)X-2+3i=0$$

, cuyas soluciones son $\dfrac{-1\pm\sqrt{3}}{2}$ y $\dfrac{1-5i}{2}$.

Ejemplo 6.2: Sea

$$P(X)=X^6+(9+i)X^5+(31-i)X^4+(45-15i)X^3+(-5-29i)X^2+(-99+5i)X-90+75$$

Para resolver la ecuación $P(X)=0$, al polinomio hay que escribirlo en la forma siguiente:

$$P(X)=X^6+9X^5+31X^4+45X^3-5X^2-99X-90+i\left(X^5-X^4-15X^3-29X^2+5X+75\right).$$

El Máximo común divisor de los polinomios reales $P_1(X)=X^6+9X^5+31X^4+45X^3-5X^2-99X-90$ y $P_2(X)=X^5-X^4-15X^3-29X^2+5X+75$ es $M(X)=X^4+4X^3+5X^2-4X-15$. Los ceros de $M(X)$ son ceros de $P(X)$ y se calcularán con el método de Lobachevski:

$$x_1 \approx -2.3027756377\,32 \qquad x_2 \approx 1.3027756377\,32$$
$$x_3 \approx -1.5+1.6583123951\,78i \qquad x_4 \approx -1.5-1.6583123951\,78i$$

Para hallar los otros do ceros complejo, se divide al polinomio $P(X)$ entre $M(X)$, utilizando la función DivExPEG obteniendo el cociente $X^2+(5+i)X+6-5i$, cuyos ceros se obtienen con la fórmula para resolver ecuaciones de segundo grado:

$$x_{5,6}=\frac{-5-i+\sqrt{30i}}{2}=\frac{-5+\sqrt{15}+\left(-1\pm\sqrt{15}\right)i}{2}$$

Ejemplo 6.3: En el caso del polinomio

$$P(X)=(3-5i)X^5+(11+4i)X^4+(-8+13i)X^3+(1-7i)X^2+(4-3i)X+19-41i,$$
$$P(X)=P_1(X)+P_2(X)$$
$$P_1(X)=3X^5+11X^4-8X^3+X^2+4X+19$$
$$P_2(X)=-5X^5+4X^4+13X^3-7X^2-3X-41$$

, y $M(X)=1$. Por tanto el polinomio no tiene ceros reales y así hay dificultades para calcular los ceros complejos. La mayoría de los polinomios se encuentra en esta situación, pero por lo menos se ha obtenido la información de que no hay ceros reales. Se podría examinar si tiene ceros imaginarios puros de la forma $X=iY$, buscando los ceros reales del polinomio

$$Q(Y)=P(iY)=(5+3i)Y^5+(11+4i)Y^4+(13+8i)Y^3+(-1+7i)Y^2+(3+4i)Y+19-41i$$

Obviamente $Q(Y)=Q_1(Y)+Q_2(Y)$, donde

$$Q_1(Y)=5Y^5+11X^4+13Y^3-Y^2+3Y+19 \quad \text{y} \quad Q_2(Y)=3Y^5+4Y^4+8Y^3+7Y^2+4X-41.$$

Con la función CalculoMCDPOL se puede establecer que el máximo común divisor de $Q_1(Y)$ *y* $Q_2(Y)$ es 1. Por tanto el polinomio $P(X)$ no tiene tampoco ceros imaginarios puros.

Ejemplo 6.4: Si es posible, simplificar la fracción siguiente:

$$R(X)=\frac{A(X)}{B(X)}=\frac{(7-11i)X^4+(43+28i)X^3+(16+7i)X^2+(33-58i)X+3+68i}{(-5+9i)X^4+(-35-18i)X^3+(-20-3i)X^2+(-29+52i)X-5-38i}$$

Utilizando la función McdPZiA, se obtiene que el máximo común divisor del numerador y del denominador de la fracción es

$$D(X)=(1-i)X^3+(4+3i)X^2+(5+i)X+8-7i$$

Luego, sabiendo que $D(X)$ es divisor de los polinomios $A(X)$ y $B(X)$, se dividen tanto a $A(X)$ como a $B(X)$ entre $D(X)$, utilizando la función DivExPEG:

$$A(X)/D(X)=(9-2i)X-4+5i \qquad B(X)/D(X)=(-7+2i)X-4+5i$$

Así,

$$R(X)=\frac{(9-2i)X-4+5i}{(-7+2i)X+2-3i}$$

Ejemplo 6.5: Sea

$$P(X)=a_0X^6+a_1X^5+a_2X^4+a_3X^3+a_4X^2+a_5X+a_6$$

, donde

$$a_0=-1333-7649i,\ a_1=32597+16136i,\ a_2=66192-74954i$$
$$a_3=200429+143122i,\ a_4=-24607+316849i,\ a_5=146692+185806i,\ a_6=47598-9086i$$

Luego,

$$P'(X)=b_0X^5+b_1X^4+b_2X^3+b_3X^2+b_4X+b_5$$

, donde

$$b_0=-7998-105894i,\ b_1=162985+80680i,\ b_2=264768-299816i$$
$$b_3=601287+429366i,\ b_4=-49214+433698i,\ b_5=146692+185806i$$

En este caso, para calcular el máximo común divisor del polinomio y de su derivada no es conveniente

utilizar las funciones que trabajan con variables de doble precisión puesto que los cálculos conducen a enteros largos y darían el resultado falso que son primos entre si. Empleando la función McdPZiBG , resulta que

$$T(X) = Mcd[P(X).\,,\, P'(X)] = (59-46i)X^2 + 155iX + 61 + 23i\,.$$

Así $P(X)$ es múltiplo de

$$[T(X)]^2 = (1365-5428i)X^4 + (14260+18290i)X^3 + (-14711-2898i)X^2 + (-7130+18910i)X + 3192 + 2806i$$

Calculando el cociente $S(X)$ entre $P(X)$ y $[T(X)]^2$ con la función DivEspPCEGG, se obtiene que:

$$P(X)/[T(X)]^2 = \alpha \cdot S[X] = \alpha \cdot \left[(1+i)X^2 + 3X + 5 + i\right] \qquad \alpha \in Z[i]$$

Así,

$$P(X) = 0 \quad \Leftrightarrow \quad T(X) = 0 \;\; ó \;\; S(X) = 0$$

Las soluciones de la ecuación $T(X)=0$ serán soluciones dobles para la ecuación $P(X)=0$, mientras que las soluciones de $S(X)=0$ serán simples.

Observación 6.1: En el caso de los polinomios de tipo (6.1), se pueden hallar siempre todos los ceros reales o imaginarios puros. La existencia de los ceros reales o ceros complejos múltiples puede aiudar en hallar todos o parte de los ceros complejos.

Observación 6.2: Si se quiere resolver la ecuación $P(X)=0$, donde

$$P(X) = (a_0 + b_0 i)X^n + (a_1 + b_1 i)X^{n-1} + \cdots + (a_{n-1} + b_{n-1} i)X + a_n + b_n i$$

, y los números $a_i \; y \; b_i \; (i = 0, \cdots, n)$ son decimales, se puede conseguir una ecuación equivalente, multiplicándola con una potencia de diez. Luego los razonamientos son como en los ejemplos 6.1 a 6.5.

Observación 6.3: Para hallar los ceros reales del polinomio

$$P(X) = (a_0 + b_0\varepsilon)X^n + (a_1 + b_1\varepsilon)X^{n-1} + \cdots + (a_{n-1} + b_{n-1}\varepsilon)X + a_n + b_n\varepsilon = 0 \qquad a_i, b_i \in Z \qquad (6.3)$$

, con coeficientes duales, $(\varepsilon \neq 0 \; y \; \varepsilon^2 = 0)$ se puede razonar de la misma manera que en el caso de los coeficientes que son enteros de Gauss.

Ejemplo 6.6: Para resolver la ecuación $P(X)=0$, donde

$$P(X) = X^3 + (-2-2\varepsilon)X^2 + (-31+6\varepsilon)X - 28 + 56\varepsilon = P_1(X) + \varepsilon\, P_2(X)$$

, se obtiene que $P_1(X) = X^3 - 2X^2 - 31X - 28$ y que $P_2(X) = -2X^2 + 6X + 56$. Teniendo en cuenta que el máximo común divisor de $P_1(X)$ y $P_2(X)$ es $M(X) = X^2 - 3X - 28$, los ceros reales de la ecuación $P(X)=0$ son soluciones de la ecuación: $X^2 - 3X - 28 = 0$ de donde se obtiene que los ceros reales son: $X_1 = 7$ y $X_2 = -4$.

Aplicando dos veces la regla de Ruffini, resulta que

$$P(X) = (X-7)\cdot\left[X^2 + (5-2\varepsilon)X + 4 - 8\varepsilon\right] = (X-7)\cdot(X+4)\cdot(X+1-2\varepsilon).$$

Así, la ecuación tiene también el cero dual. $X_3 = -1 + 2\varepsilon$

Bibliografia:

[1] Ion D.Ion, Nicolae Radu, Algebra, Editura Didactică ș i Pedagogică, Bucureș ti, 1970.
[2] A.I. Froda, Algebra superioară, Editura Didactică ș i Pedagogică, Bucureș ti, 1958.
[3] Aladar Peter Santha, Cálculos con números enteros largos en ordenadores, Monografías.com, 2012
[4] Aladar Peter Santha, Costrucción y utilización de las series de polinomios Sturm en un ordenador, Monografias.com, 2012.