GIIT_AMA_2223_Practica4_alokenveo_jelanang

last edited Nov 16, 2022, 2:39:16 PM by GIT21jelanang

Práctica 4. Introducción a la Teoría de Números: Ecuaciones Diofánticas

Temporalización de la práctica: Lunes 7 de noviembre y Lunes 14 de noviembre de 2022

Entrega de la práctica: Desde el 14 de noviembre hasta el 21 de noviembre (ver tarea en el campus virtual o agenda de la asignatura)

Instrucciones:

1. Haz una copia de la hoja pública y renómbrala: si tu correo es **mariomp@alumnos.unex.es** añade al final del título **_mariomp**, por ejemplo GIIT_AMA_2223_Practica4_mariomp

- Para cambiar el nombre pulsa en el título de la hoja (arriba del todo, entre el logo de Sage y el menú "Archivo/File...")

2. Comparte la hoja de trabajo con el usuario **mariomp2223** mediante el botón Compartir/Share de arriba a la derecha. Si lo hacéis en pareja, compartid la hoja con el usuario de tu compañero (ambos usuarios separados por comas), así como añadir también el usuario en el título de la hoja (separados por _).

3. Completa la primera celda y trabaja la práctica.

4. Cuando hayas terminado, haz una copia en un único fichero PDF y ponlo en el campus virtual (Si lo hacéis en pareja, basta que lo suba uno). **Esa será la versión que se evaluará**. La hoja no se considera entregada si no se ha renombrado y compartido (pasos 1 y 2).

- Para generar el PDF lo más sencillo es usar el botón Imprimir/Print de arriba e imprimir la nueva página a fichero.

5. Una vez subido el PDF al campus virtual, no podrá modificarse esta hoja de trabajo. **Hacerlo conllevará la calificación de 0 en esta práctica**.

6. Los ejercicios a entregar se representan en Rojo y deben estar correctamente explicados. Los ejercicios indicados con ** **NO** serán obligatorios, pero aquellos que los hagan podrán ir sumando por cada uno de ellos 0.1 puntos adicionales en la calificación de la práctica.

Alumno/s: Alfredo Mituy Okenve, José Luis Obiang Ela Nanguan

La Teoría de Números surgió al tener que resolver problemas con números enteros. Con la computación, esta teoría tiene una gran dimensión práctica, pues en última instancia los ordenador solo trabajan con números enteros, y que puede aplicarse en: criptografía, códigos correctores y detectores de errores, procesado de señales digitales,...

División Entera

Sage permite operar con números enteros sobre los que se pueden aplicar multitud de operaciones, por ejemplo se puede determinar, de modo sencillo, el resto de la división de dos números enteros a y b, con el símbolo **%**. Por ejemplo, veamos que efectivamente el resto de dividir 7 entre 2 es 1.

\n
\n

\n

\n

\n\n\n\n\n
' });

```
7%2
```

```
1
```

Algoritmo de la División

Sean $a, b \in \mathbb{Z}, b \neq 0$, existen $c, r \in \mathbb{Z}$ únicos tales que

$$a = bc + r, 0 \leq r < |b|$$

donde:

- a es el dividendo
- b es el divisor
- c es el cociente
- r es el resto

Con lo visto anteriormente, se puede calcular r como $a\%b$, y el cociente se calcula con el símbolo //, esto es, a//b es el cociente de dividir a entre b.

```
a, b = 7,2
c = a//b
r = a%b
print "Dividendo: ", a
print "Divisor: ", b
print "Cociente: ", c
print "Resto: ", r
```

```
Dividendo:  7
Divisor:  2
Cociente:  3
Resto:  1
```

Veamos que efectivamente se cumple el algoritmo de la división.

```
b*c + r
```

```
7
```

Además, también podemos trabajar con números negativos de manera muy sencilla.

```
a, b = -26,5
c = a//b
r = a%b
print "Dividendo: ", a
print "Divisor: ", b
print "Cociente: ", c
print "Resto: ", r
print b*c + r
```

```
Dividendo:  -26
Divisor:  5
Cociente:  -6
Resto:  4
-26
```

Divisibilidad

Por otro lado, recordemos que se dice que b es ***divisor*** de a si el resto de dividir a entre b es 0, o equivalentemente se dice que a es ***múltiplo*** de b. Luego, con Sage es sencillo comprobar si un número es divisor de otro, pues basta determinar el resto de la división. Por ejemplo, comprobemos que 169 es múltiplo de 13 (o que 13 es divisor de 169).

```
a,b = 169,13
a%b
```

```
0
```

De manera general, Sage permite conocer todos los divisores de un número entero n empleando la función ***divisors(n)*** que divide una lista con todos ellos. Por ejemplo, determinemos los divisores de 91.

```
divisors(91)
```

```
[1, 7, 13, 91]
```

Para obtener los múltiplos de un número es necesario recordar qué signfica, esto es, se dice que a es múltiplo de b si

$$a = bd$$

o equivalentemente si a. Por ejemplo, veamos cómo determinar los múltiplos de 3 (solo mostraremos aquellos que son menores que 100).

```
b = 3
multiplos = [a for a in [1..100] if (a%b)==0]
print multiplos
```

```
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54,
57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99]
```

Pero existe otro modo mucho más sencillo, ya que los múltiplos de 3 son aquellos de la forma $3c$, por lo que podemos construir la lista de modo más sencillo.

```
b = 3
multiplos = [b*c for c in [1..100]]
print multiplos
```

```
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54,
57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105,
108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147,
150, 153, 156, 159, 162, 165, 168, 171, 174, 177, 180, 183, 186, 189,
192, 195, 198, 201, 204, 207, 210, 213, 216, 219, 222, 225, 228, 231,
234, 237, 240, 243, 246, 249, 252, 255, 258, 261, 264, 267, 270, 273,
276, 279, 282, 285, 288, 291, 294, 297, 300]
```

Por otra parte, Sage permite comprobar si un número n es primo o no con la función ***is_prime(n)***.

```
n = 7
is_prime(n)
```

```
True
```

También podemos calcular el máximo común divisor y el mínimo común múltiplo de dos números enteros a y b a partir de las funciones ***gcd(a,b)*** y ***lcm(a,b)***. Por ejemplo, veamos cual es el mcd y el mcm de los números 6 y 15.

```
a,b = 6,15
print "mcd = ", gcd(a,b)
print "mcm = ", lcm(a,b)
```

```
mcd =  3
mcm =  30
```

Para comprobar si dos números son primos entre sí es necesario ver si mcd(a,b) = 1. Por lo que bastará determinar el máximo común divisor de dos números enteros a y b, para comprobar si son primos entre sí o no.

```
a,b = 6,35
gcd(a,b)
```

```
1
```

Observamos que 6 y 35 sí son primos entre sí.

Identidad de Bézout

Sean $a, b \in \mathbb{Z}, a, b \neq 0$. El máximo común divisor (d) de a y b es el entero positivo más pequeño que puede expresarse en la forma

$$d = ax + by$$

para algún $x, y \in \mathbb{Z}$.

En particular, si a y b son primos entre sí si y solo si existen $x, y \in \mathbb{Z}$ tales que

$$ax + by = 1$$

Para poder determinar los enteros x, y de la Identidad de Bézout es necesario aplicar el Algoritmo de Euclides.

Algoritmo de Euclides

Sean $a, b \neq 0$ números enteros. Sea r tal que $a = bc + r$. Entonces:

- Si $r = 0$, entonces $mcd(a, b) = 1$
- Si $r \neq 0$, entonces $mcd(a, b) = mcd(b, r)$.

De modo que este algoritmo permite determinar el máximo común divisor entre dos enteros a y b. Este algoritmo es iterativo, que parará cuando el resto sea 0. Por ejemplo, determinemos el máximo común divisor de 312 y 120 utilizando el algoritmo de Euclides.

```
a,b = 312,120
r = a%b
print "Resto: ", r
```

```
Resto:  72
```

Como $r = 72$, entonces $mcd(312, 120) = mcd(120, 72)$, por lo que hay que repetir el proceso.

```
a,b = b,r
r = a%b
print "Resto: ", r
```

```
Resto:  48
```

Como $r \neq 0$, $mcd(120, 72) = mcd(72, 48)$.

```
a,b = b,r
r = a%b
print "Resto: ", r
```

```
Resto:  24
```

Como $r \neq 0$, $mcd(72, 48) = mcd(48, 24)$.

```
a,b = 120,r
r = a%b
print "Resto: ", r
```

```
Resto:  0
```

Acabamos de llegar a $r = 0$, de manera que mcd(312,120) = mcd(48,24) = 24.

Ejercicio 1. Crea una función que aplique el Algoritmo de Euclides para determinar el máximo común divisor de dos números enteros a y b, mostrando en cada caso el algoritmo de la División (a = bc + r), esto es, debe mostrar tanto el cociente como el resto en cada paso.

```
#Realiza el ejercicio en esta celda
import numpy as np
def getMaximoComunDivisor(a,b,tipo):
    mcd = 0
    if a < b:    #Hacemos el intercambio en caso de que b(divisor) sea mayor que a(dividendo)
        d = a; a = b; b = d
    d = a; e = b   #Hacemos una copia del dividendo y divisor antes de empezar con las modificaciones
    lista = []    #En esta lista almacenaremos los cocientes, restos y mcd(a,b)
    cocientes = []
    restos = []
    if a != 0 and b != 0:    #Comprobamos si a o b son iguales a 0, en caso contrario se ejecuta el programa, sino
muestra que no existe MCD
        r=1
        while r!= 0:
            c=a//b   #Obtenemos el cociente
            r=a%b    #Obtenemos el resto
            if r != 0:
                mcd = r
            else:
                mcd=b
            if tipo == 2:    #Si queremos mostrar a=b*c+r
                print a,"=",b,"*",c,"+",r
            elif tipo == 3:
                cocientes.append(c)
                restos.append(r)
            a,b=b,r


        if tipo == 2:    #Si queremos mostrar solo el maximo comun divisor
            print "mcd(",d,",",e,")=",mcd
        elif tipo == 3:    #Si queremos mostrar la lista de cocientes, resto y MCD
            lista.append([cocientes,restos,mcd])
            return lista
    else:
        print "No existe Maximo Comun Divisor"
```

Ejercicio 2. Aplica la función anterior para determinar el máximo común divisor de 123456 y 789012.

```
#Realiza el ejercicio en esta celda
tipo = 2    #Asignamos a la variable tipo valor 2 para mostrar a=b*c+r
if getMaximoComunDivisor(123456,789012,tipo)==false:
    getMaximoComunDivisor(123456,789012,tipo)
```

```
789012 = 123456 * 6 + 48276
123456 = 48276 * 2 + 26904
48276 = 26904 * 1 + 21372
26904 = 21372 * 1 + 5532
21372 = 5532 * 3 + 4776
5532 = 4776 * 1 + 756
4776 = 756 * 6 + 240
756 = 240 * 3 + 36
240 = 36 * 6 + 24
36 = 24 * 1 + 12
24 = 12 * 2 + 0
mcd( 789012 , 123456 )= 12
```

Ejercicio 3. Modifica la función del ejercicio 1, para que en lugar de mostrar por pantalla el algoritmo de la División, devuelva una lista formada por 3 elementos (en el mismo orden): una lista con los cocientes, una lista con los restos, y el máximo común divisor.

```
#Realiza el ejercicio en esta celda

def getListaMCD(a,b, tipo):
    print "De izquierda a derecha tenemos los diferentes cocientes, los restos, y el maximo comun divisor"
    show(getMaximoComunDivisor(a,b,tipo))    #Nos devuelve la lista de cocientes, restos y mcd siempre y cuando el
parametro tipo sea distinto de 2
```

Ejercicio. 4. Aplica la función del ejercicio anterior para los números enteros 123456 y 789012.

```
#Realiza el ejercicio en esta celda
tipo = 3    #Asignamos a la variable tipo valor 3 para mostrar únicamente la lista de cocientes, restos y mcd.
getListaMCD(123456,789012,tipo)
```

```
De izquierda a derecha tenemos los diferentes cocientes, los restos, y
el maximo comun divisor
```

$[[[6, 2, 1, 1, 3, 1, 6, 3, 6, 1, 2], [48276, 26904, 21372, 5532, 4776, 756, 240, 36, 24, 12, 0], 12]]$

El algoritmo de Euclides permite encontrar los enteros x e y de la Identidad de Bézout

$$ax + by = mcd(a, b)$$

Por ejemplo, busquemos los enteros tales que se cumple la Identidad de Bézout para los números 312 y 120, esto es,

$$312x + 120y = 24$$

Para ello, hemos visto que el Algoritmo de Euclides mostraba los siguientes resultados:

$$312 = 120 \cdot 2 + 72$$

$$120 = 72 \cdot 1 + 48$$

$$72 = 48 \cdot 1 + 24$$

$$48 = 24 \cdot 2 + 0$$

Si despejamos los restos en cada caso:

$$72 = 312 - 2 \cdot 120$$

$$48 = 120 - 1 \cdot 72$$

$$24 = 72 - 1 \cdot 48$$

y a partir de la última ecuación, podemos ir sustituyendo de forma regresiva. Esto es:

$$24 = 72 - 1 \cdot 48 = 72 - 1 \cdot (120 - 1 \cdot 72) =$$

$$= 72 - 1 \cdot 120 + 1 \cdot 72 = 2 \cdot 72 - 1 \cdot 120 =$$

$$= 2 \cdot (312 - 2 \cdot 120) - 1 \cdot 120 = 2 \cdot 312 - 4 \cdot 120 - 1 \cdot 120 =$$

$$= 312 \cdot 2 + 120 \cdot (-5)$$

Por lo tanto,

$$24 = 312 \cdot 2 + 120 \cdot (-5)$$

esto es, los números enteros buscados son $x = 2, y = -5$. Este proceso se conoce como ***Algoritmo de Euclides Extendido***.

Ejercicio 5. Crea una función que aplique el Algoritmo de Euclides Extendido para encontrar los enteros x, y de la Identidad de Bézout de dos números a, b. Esto es, la función debe recibir dos enteros a,b y devolver otros dos enteros x,y.

```
#Realiza el ejercicio en esta celda
def EuclidesExtend(a,b):
    a_ini=a
    b_ini=b
    tipo=5    #Tipo distinto de 2 y 3
    d=getMaximoComunDivisor(a,b,tipo)    #Obtenemos el mcd utilizando la función getMaximoComunDivisor creada
anteriormente pasandole un tipo distinto del 2 y 3, devuelve únicamente el mcd

    u0 = 1
    u1 = 0
    v0 = 0
    v1 = 1

    print "Mostramos la Identidad de Bézout: x,y: ",a,"* x +",b,"* y =",d
    print "Algoritmo de Euclides Extendido"
    r=1
    while r>0:
        c=a//b
        r=a - b * c
        u = u0 - c * u1      #Hacemos sustituciones sucesivas de los restos de cada una de las ecuaciones obtenidas
en el algoritmo de Euclides Extendido
        v = v0 - c * v1
        print a,"=",b,"*",c,"+",r

        #Actualizamos a,b
        a=b
        b=r
        #Actualizamos la siguiente iteración
        u0 = u1
        u1 = u
        v0 = v1
        v1 = v

    r=1
    print "\nDespejamos los restos:"
    while r>0:
        c=a_ini//b_ini
        r=a_ini%b_ini
        if r > 0:
            print r,"=",a_ini,"-",b_ini,"*",c

        a_ini=b_ini; b_ini=r

    return u0,v0   #Devolvemos las soluciones particulares (x, y)



EuclidesExtend(312,120)
```

```
Mostramos la Identidad de Bézout: x,y:  312 * x + 120 * y = 24
Algoritmo de Euclides Extendido
312 = 120 * 2 + 72
120 = 72 * 1 + 48
72 = 48 * 1 + 24
48 = 24 * 2 + 0

Despejamos los restos:
72 = 312 - 120 * 2
48 = 120 - 72 * 1
24 = 72 - 48 * 1
(2, -5)
```

Ejercicio 6. Aplica la función anterior para resolver la Identidad de Bézout de los enteros 124 y 650.

```
#Realiza el ejercicio en esta celda
EuclidesExtend(124,650)
```

```
Mostramos la Identidad de Bézout: x,y:  124 * x + 650 * y = 2
Algoritmo de Euclides Extendido
124 = 650 * 0 + 124
650 = 124 * 5 + 30
124 = 30 * 4 + 4
30 = 4 * 7 + 2
4 = 2 * 2 + 0

Despejamos los restos:
124 = 124 - 650 * 0
30 = 650 - 124 * 5
4 = 124 - 30 * 4
```

```
2 = 30 - 4 * 7
(-152, 29)
```

Ecuaciones Diofánticas

Una ecuación diofántica es una ecuación algebraica para la cual se buscan soluciones enteras. Las más sencillas son las ecuaciones diofánticas lineales: dados $a, b, n \in \mathbb{Z}$, se trata de encontrar los enteros x, y tales que

$$ax + by = n$$

Para poder resolver la ecuación diofántica es necesario comprobar, previamente, si existe solución entera. Para ello, la ecuación $ax + by = n$ tiene solución entera si y solo si n es múltiplo de mcd(a,b), esto es, $n = m \cdot mcd(a, b)$.

Si la ecuación diofántica tiene solución es fácil resolverla, puesto que si consideramos la Identidad de Bézout:

$$a\bar{x} + b\bar{y} = mcd(a, b)$$

es equivalente a

$$a \cdot m\bar{x} + b \cdot m\bar{y} = m \cdot mcd(a, b)$$

y denotando por $x = m\bar{x}, y = m\bar{y}$ obtenemos la ecuación diófantica

$$ax + by = n$$

Por lo tanto, para resolver la ecuación diofántica, en primer lugar es necesario comprobar que existe solución entera, y en caso afirmativo:

1) Se resuelve la Identidad de Bézout: $\bar{x}, \bar{y}$

2) Obteniendo el cociente entero entre n y mcd(a,b): m

3) Se resuelve la ecuación diofántica: $x = m\bar{x}, y = m\bar{y}$

Por ejemplo, se quiere resolver la ecuación diofántica

$$312x + 120y = 168$$

Para comprobar si tiene solución bastará ver si el resto de 168 y el mcd(312,120)=24 es 0.

```
168%24
```

```
0
```

Efectivamente, la ecuación diofántica tiene solución, resolvamos entonces la Identidad de Bézout $312\bar{x} + 120\bar{y} = mcd(312, 120) = 24$, que ya hemos visto que las soluciones son $\bar{x} = 2, \bar{y} = -5$. Una vez resuelta dicha identidad (que puede verse que se trata también de una ecuación diofántica), se debe determinar el cociente entero entre 168 y mcd(312,120) = 24.

```
m = 168//24
print "Cociente: ", m
```

```
Cociente:  7
```

Por tanto, las soluciones de la ecuación diofántica

$$312x + 120y = 168$$

son $x = 7\bar{x}, y = 7\bar{y}$, luego

```
x = m*2
y = m*(-5)
print "Soluciones de la Ecuación Diofántica: x = ",x,"y = ",y
```

```
Soluciones de la Ecuación Diofántica: x =  14 y =  -35
```

Comprobemos que efectivamente son soluciones:

```
312*x+120*y
```

```
168
```

Sin embargo, otra característica especial que tienen las ecuaciones diofánticas es que las soluciones no son únicas, en concreto las soluciones anteriores se conocen como ***Soluciones Particulares*** de la ecuación diofántica x_0, y_0.

```
x0,y0 = 14,-35
```

El resto de soluciones se obtienen a partir de la ***Soluciones Generales*** de la ecuación diofántica, definidas como:

$$x = x_0 + k\frac{b}{mcd(a,b)}, y = y_0 - k\frac{a}{mcd(a,b)}, k \in \mathbb{Z}$$

Mostremos algunas soluciones con valores de $k = -2, -1, 0, 1, 2$. Muy importante es tener en cuenta que para un mismo valor de k se tienen las soluciones x, y de la ecuación diofántica. Si nos fijamos, además, para $k = 0$ se obtienen las soluciones particulares.

```
a,b = 312,120
mcd = gcd(a,b)
x0,y0 = 14,-35
x = [x0 + k*b/mcd for k in [-2..2]]
y = [y0 - k*a/mcd for k in [-2..2]]
for k in [-2..2]:
    print "Solución para k = ",k,": x = ",x[k+2],", y = ",y[k+2]
```

```
Solución para k =  -2 : x =  4 , y =  -9
Solución para k =  -1 : x =  9 , y =  -22
Solución para k =  0 : x =  14 , y =  -35
Solución para k =  1 : x =  19 , y =  -48
Solución para k =  2 : x =  24 , y =  -61
```

Ejercicio 7. Resuelve las siguientes ecuaciones diofánticas:

a) $24x + 14y = 30$

b) $4x + 12y = 13$

```
#Realiza el ejercicio en esta celda
def resolvEcDiofantica(a,b,n,minx, maxx, miny, maxy, tipo):
    mcd = getMaximoComunDivisor(a,b,tipo)
    if tipo == 12:
        print "Trabajaremos en centavos para evitar operar con decimales\n"
        print "Definimos dos variables: a-->dolares y b->centavos\n"
        print "Valor del cheque-> 100a+b\n"
        print "Pero como en el banco le cambiaron los papeles de dolares y centavos\n"
        print "El señor Martinez recibió 100b+a-68\n"
        print "La anterior cantidad nos dicen que equivale al doble del importe del cheque: 100b+a-68=200a+2b"
        print "\nNos queda la ecuacion 199a-98b=-68"
        print "\nProcedemos a obtener el mcd(a.b) utilizando el algoritmo de Euclides"

    #Primero comprobamos si existe solución
    print "Comprobamos si existe solución"    #Existe solucion si n es multiplo de mcd entre a y b
    if n % mcd == 0:
        print "****************Hay solución****************"
        x0,y0=EuclidesExtend(a,b)    #Reutilizamos la función EuclidesExtend() para obtener las soluciones
particulares
        if tipo != 12:
            print "Soluciones particulares x,y: ","(",x0,",",y0,")"
            m = n//mcd
            x1 = m*x0
            y1 = m*y0

            #Hallamos las soluciones globales
            print "Soluciones globales"
            print "x =",x1," +","k *",b/mcd
            print "y =",y1," +","k *",a/mcd
        else:
            print "Despejamos la b: b=(199a+68)/98\n"
            print "Descomponemos 199a en (196+3)a y entonces simplificamos 196a dividiendolo entre 98"
            print "\nObtenemos b=2a+(3a+68)/98 la cual será igual a 2a+b'\n"
            print "Sustituimos (3a+68)/98 por b'-> b'=(3a+68)/98\n"
            print "Despejamos a de la ecuacion anterior-> a=(98b'-68)/3\n"
            print "Ahora procedemos a dar valores a b' de restos modulo 3(0,1,2)"
            print "\nNos quedamos con el valor de b' que haga que al sustituirlo en la ecuación de a nos de un valor
positivo y el menor valor de a\n"
            b1=0
            b2=var('b2')
            f(b2)=(98*b2-68)/3
            positivos=[]
            #Añadimos en una lista solo los valores de b que hacen que a sea positivo
            for i in range(0,3):
                if f(i) > 0:
                    positivos.append(i)

            menor = f(positivos[0])    #Asumimos que menor es el primer valor de a
            for i in positivos:
                if f(i) <= menor:
                    menor = f(i)
                    b1=i
            print "El valor de b' es",b1," y por lo tanto, el valor de a es:",menor
            a2 = var('a2')
            f1(a2)=2*a2+(3*a2+68)/98
            print "\n Entonces el valor de b es:", f1(menor)
            print "\nMientras que la solución general es:"
            print "\nObtenemos la siguiente ecuacion diofantica"
            print "\na =",menor," - 98t"
            print "\nb =",f1(menor)," - 199t"
            print "\nEl valor mínimo se alcanza cuando t = 0, donde a =", menor," y b =",f1(menor)
            print "\nAsí el Señor Martínez viajó a Estados Unidos con un cheque de viaje cuyo valor ascendía a 10
dólares y 21 centavos."

        if tipo == 8:    #En caso de que queramos mostrar también los distintos valores que toman las soluciones
globales cuando modificamos el valor de k
            x = [x1 + k*b/mcd for k in [-2..2] ]
            y = [y1 - k*a/mcd for k in [-2..2]]
            for k in [-2..2]:
               if x1 + k*b/mcd >= minx and x1 + k*b/mcd <= maxx:
                   print "Solución para k = ",k,": x = ",x1 + k*b/mcd
               if y1 + k*b/mcd >= miny and y1 + k*b/mcd <= maxy:
                   print "Solución para k = ",k,": y = ",y1 + k*b/mcd

    else:
        print "****************No existe solución****************"

#a)
```

```
Comprobamos si existe solución
****************Hay solución****************
```

```
Mostramos la Identidad de Bézout: x,y:  24 * x + 14 * y = 2
Algoritmo de Euclides Extendido
24 = 14 * 1 + 10
14 = 10 * 1 + 4
10 = 4 * 2 + 2
4 = 2 * 2 + 0

Despejamos los restos:
10 = 24 - 14 * 1
4 = 14 - 10 * 1
2 = 10 - 4 * 2
Soluciones particulares x,y:  ( 3 , -5 )
Soluciones globales
x = 45  + k * 7
y = -75  + k * 12
```

```
#b)
a=4; b=12; n=13
resolvEcDiofantica(a,b,n,0,0,0,0,1)
```

```
Comprobamos si existe solución
***************No existe solución***************
```

Ejercicio 8. Resuelve la siguiente ecuación diofántica:

$$452x + 284y = 440$$

Encuentra las soluciones que cumplan que $2300 \leq x \leq 2600, -4100 \leq y \leq -3700$.

```
#Realiza el ejercicio en esta celda
a=452; b=284; n=440
resolvEcDiofantica(a,b,n, 2300,2600,-4100,-3700,8)
```

```
Comprobamos si existe solución
***************Hay solución***************
Mostramos la Identidad de Bézout: x,y:  452 * x + 284 * y = 4
Algoritmo de Euclides Extendido
452 = 284 * 1 + 168
284 = 168 * 1 + 116
168 = 116 * 1 + 52
116 = 52 * 2 + 12
52 = 12 * 4 + 4
12 = 4 * 3 + 0

Despejamos los restos:
168 = 452 - 284 * 1
116 = 284 - 168 * 1
52 = 168 - 116 * 1
12 = 116 - 52 * 2
4 = 52 - 12 * 4
Soluciones particulares x,y:  ( 22 , -35 )
Soluciones globales
x = 2420  + k * 71
y = -3850  + k * 113
Solución para k =  -2 : y =  -3992
Solución para k =  -1 : x =  2349
Solución para k =  -1 : y =  -3921
Solución para k =  0 : x =  2420
Solución para k =  0 : y =  -3850
Solución para k =  1 : x =  2491
Solución para k =  1 : y =  -3779
Solución para k =  2 : x =  2562
Solución para k =  2 : y =  -3708
```

Ejercicio 9. Encuentra las soluciones de la ecuación diofántica:

$$2351x + 23y = 37$$

```
#Realiza el ejercicio en esta celda
a=2351; b=23; n=37
resolvEcDiofantica(a,b,n,0,0,0,0,1)
```

```
Comprobamos si existe solución
***************Hay solución***************
Mostramos la Identidad de Bézout: x,y:  2351 * x + 23 * y = 1
Algoritmo de Euclides Extendido
```

```
2351 = 23 * 102 + 5
23 = 5 * 4 + 3
5 = 3 * 1 + 2
3 = 2 * 1 + 1
2 = 1 * 2 + 0

Despejamos los restos:
5 = 2351 - 23 * 102
3 = 23 - 5 * 4
2 = 5 - 3 * 1
1 = 3 - 2 * 1
Soluciones particulares x,y:  ( -9 , 920 )
Soluciones globales
x = -333  + k * 23
y = 34040  + k * 2351
```

Ejercicio 10. Calcula dos números enteros x, y que sean solución de la ecuación diofántica

$$2112x + 123y = 24$$

```
#Realiza el ejercicio en esta celda
a=2112; b=123; n=24
resolvEcDiofantica(a,b,n,0,0,0,0,1)
```

```
Comprobamos si existe solución
***************Hay solución***************
Mostramos la Identidad de Bézout: x,y:  2112 * x + 123 * y = 3
Algoritmo de Euclides Extendido
2112 = 123 * 17 + 21
123 = 21 * 5 + 18
21 = 18 * 1 + 3
18 = 3 * 6 + 0

Despejamos los restos:
21 = 2112 - 123 * 17
18 = 123 - 21 * 5
3 = 21 - 18 * 1
Soluciones particulares x,y:  ( 6 , -103 )
Soluciones globales
x = 48  + k * 41
y = -824  + k * 704
```

Ejercicio 11. Resuelve la siguiente ecuación diofántica

$$2110x + 120y = 20$$

```
#Realiza el ejercicio en esta celda
a=2110; b=120; n=20
resolvEcDiofantica(a,b,n,0,0,0,0,1)
```

```
Comprobamos si existe solución
***************Hay solución***************
Mostramos la Identidad de Bézout: x,y:  2110 * x + 120 * y = 10
Algoritmo de Euclides Extendido
2110 = 120 * 17 + 70
120 = 70 * 1 + 50
70 = 50 * 1 + 20
50 = 20 * 2 + 10
20 = 10 * 2 + 0

Despejamos los restos:
70 = 2110 - 120 * 17
50 = 120 - 70 * 1
20 = 70 - 50 * 1
10 = 50 - 20 * 2
Soluciones particulares x,y:  ( -5 , 88 )
Soluciones globales
x = -10  + k * 12
y = 176  + k * 211
```

Ejercicio 12. Estando en Estados Unidos, el Sr. Martínez cambió un cheque de viaje. El cajero, al pagarle, confundió el número de dólares con el de centavos y viceversa. El Sr. Martínez gastó 68 centavos en sellos y comprobó que el dinero que le quedaba era el doble del cheque de viaje que había cambiado. ¿Qué valor mínimo tenía el cheque?

```
#Realiza el ejercicio en esta celda
a=199; b=98; n=68
resolvEcDiofantica(a,b,n,0, 0, 0, 0, 12)
```

```
Trabajaremos en centavos para evitar operar con decimales

Definimos dos variables: a-->dolares y b->centavos

Valor del cheque-> 100a+b

Pero como en el banco le cambiaron los papeles de dolares y centavos

El señor Martinez recibió 100b+a-68

La anterior cantidad nos dicen que equivale al doble del importe del
cheque: 100b+a-68=200a+2b

Nos queda la ecuacion 199a-98b=-68

Procedemos a obtener el mcd(a.b) utilizando el algoritmo de Euclides
Comprobamos si existe solución
***************Hay solución***************
Mostramos la Identidad de Bézout: x,y:  199 * x + 98 * y = 1
Algoritmo de Euclides Extendido
199 = 98 * 2 + 3
98 = 3 * 32 + 2
3 = 2 * 1 + 1
2 = 1 * 2 + 0

Despejamos los restos:
3 = 199 - 98 * 2
2 = 98 - 3 * 32
1 = 3 - 2 * 1
Despejamos la b: b=(199a+68)/98

Descomponemos 199a en (196+3)a y entonces simplificamos 196a
dividiendolo entre 98

Obtenemos b=2a+(3a+68)/98 la cual será igual a 2a+b'

Sustituimos (3a+68)/98 por b'-> b'=(3a+68)/98

Despejamos a de la ecuacion anterior-> a=(98b'-68)/3

Ahora procedemos a dar valores a b' de restos modulo 3(0,1,2)

Nos quedamos con el valor de b' que haga que al sustituirlo en la
ecuación de a nos de un valor positivo y el menor valor de a

El valor de b' es 1  y por lo tanto, el valor de a es: 10

 Entonces el valor de b es: 21

Mientras que la solución general es:

Obtenemos la siguiente ecuacion diofantica

a = 10  - 98t

b = 21  - 199t

El valor mínimo se alcanza cuando t = 0, donde a = 10  y b = 21

Así el Señor Martínez viajó a Estados Unidos con un cheque de viaje cuyo
valor ascendía a 10 dólares y 21 centavos.
```

evaluate