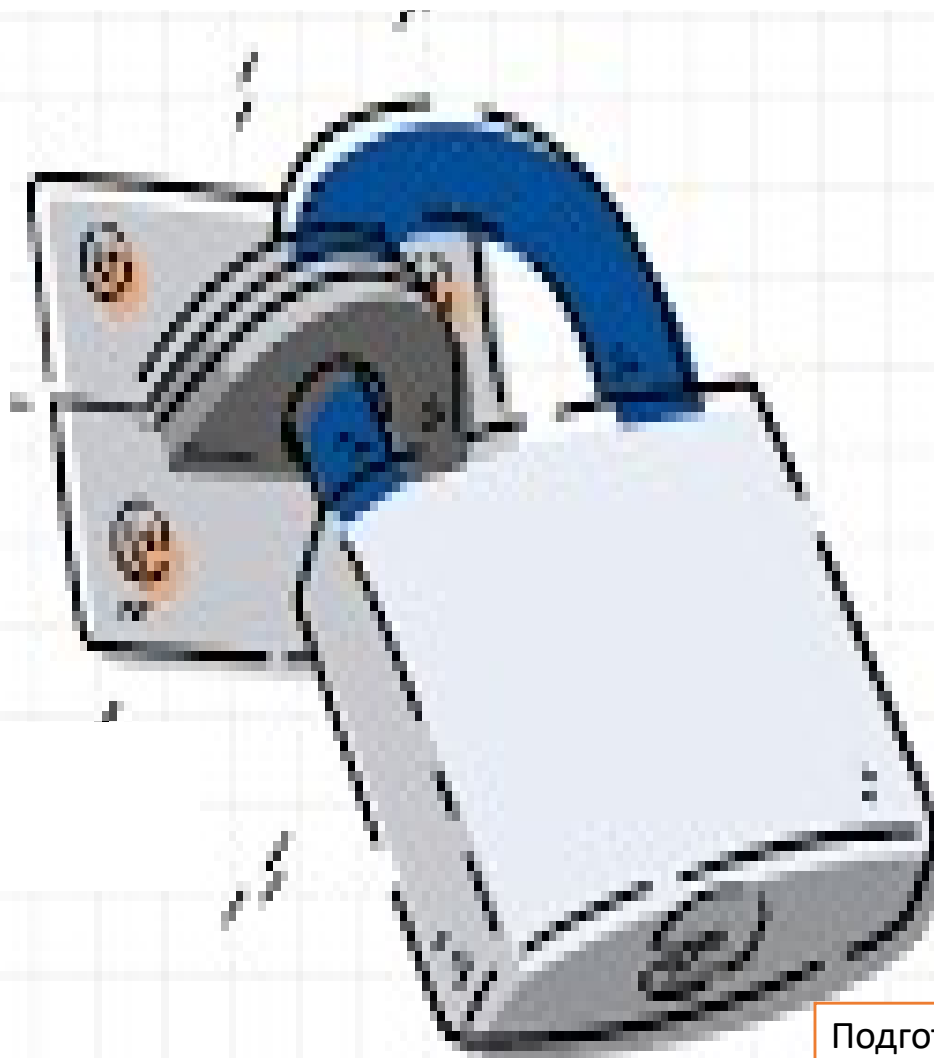


Российский университет дружбы народов Научный факультет

Математические
основы защиты
информации и
информационной
безопасности

МСД



Подготовлено студентом:
Елиенис Санчес Родригес.
Преподаватель: Дмитрий Сергеевич

МСД

В математике наибольший общий делитель или MCD называется наибольшим числом, которое делит ровно два или более чисел одновременно. Поскольку мы говорим о наибольшем числе, мы будем принимать во внимание только положительные делители.

.

МСД

Один из методов вычисления максимального общего делителя - алгоритм Евклида, он объясняет, что наибольший общий делитель двух чисел можно найти, разделив большее число на меньшее. Если деление точное, $m.c.d.$ наименьшее число. Если деление неточное, то берется остаток и делится столько раз, сколько потребуется, чтобы получить деление без остатка. $M. c. d.$ - последнее число, на которое его можно разделить.

МСД Python

```
from pyfiglet import figlet_format
print(figlet_format( "MCD by Elienis", font = "cybermedium"))
import math
```

```
comprobar = True
```

while comprobar:

```
#introducimos las variables en numeros enteros
```

```
a = int(input("ingrese el primer número:"))
```

```
b = int(input("ingrese el segundo número:"))
```

MCD = False

si a y b es mayor que 0 y que a y b sean diferentes

if $a > 0$ and $b > 0$ and $a \neq b$:

```
comprobar = False
```

```
#creo variable auxiliar en caso de que b sea menor que a
```

```
if b < a:
```

$$\text{aux} = a$$
$$a=b$$

b= aux

i=a

```
#creamos ciclo mientras mcd sea falso y i sea mayor o igual a 1
```

```
while not MCD and i >= 1:
```

```
#si a es igual a 0 e i , imprimimos esa variable
```

if $a \% i == 0$ and $b \% i == 0$:

```
print(" el MCD es ",i)
```

MCD=True

#decrementamos 1 en el bucle hasta que i sea una division exacta

```
else:
```

$$i = 1$$

```
# si el usuario no coloco los numeros correctos , manda a pedir los numeros
```

```
else:
```

```
if a == b:
```

```
print("Los numeros son iguales, intentelo de nuevo ")
```

```
else:
```

```
print("Los numeros no son correctos, intentelo neuvamente")
```

результат

The screenshot shows the PyCharm IDE interface. The top toolbar contains buttons for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The main window displays the 'Run' output for 'main.py'. The output shows a pattern of asterisks and backslashes, followed by user input for two numbers (44 and 33) and the calculated MCD (11). The bottom status bar indicates 'Version Control', 'Run', 'TODO', 'Problems', 'Terminal', 'Python Packages', 'Python Console', and 'Services'.

расширенный алгоритм Евклида

это небольшая модификация, которая позволяет дополнительно выразить наибольший общий делитель как линейную комбинацию. Этот алгоритм находит применение, в частности, в различных областях, таких как алгебра, теория чисел и информатика. С небольшими модификациями он обычно используется в электронных компьютерах из-за его высокой эффективности.

расширенный алгоритм Евклида

```
from pyfiglet import figlet_format
print(figlet_format("MCD Euclides by Elienis", font = "cybermedium"))
import math
```

```
def euclides(num1, num2, iteracciones=1):
    # Si el num1 es inferior al num2, los invertimos
    if num1 < num2:
        num1, num2 = num2, num1

    # obtenemos el resto de la division
    resto = num1 % num2

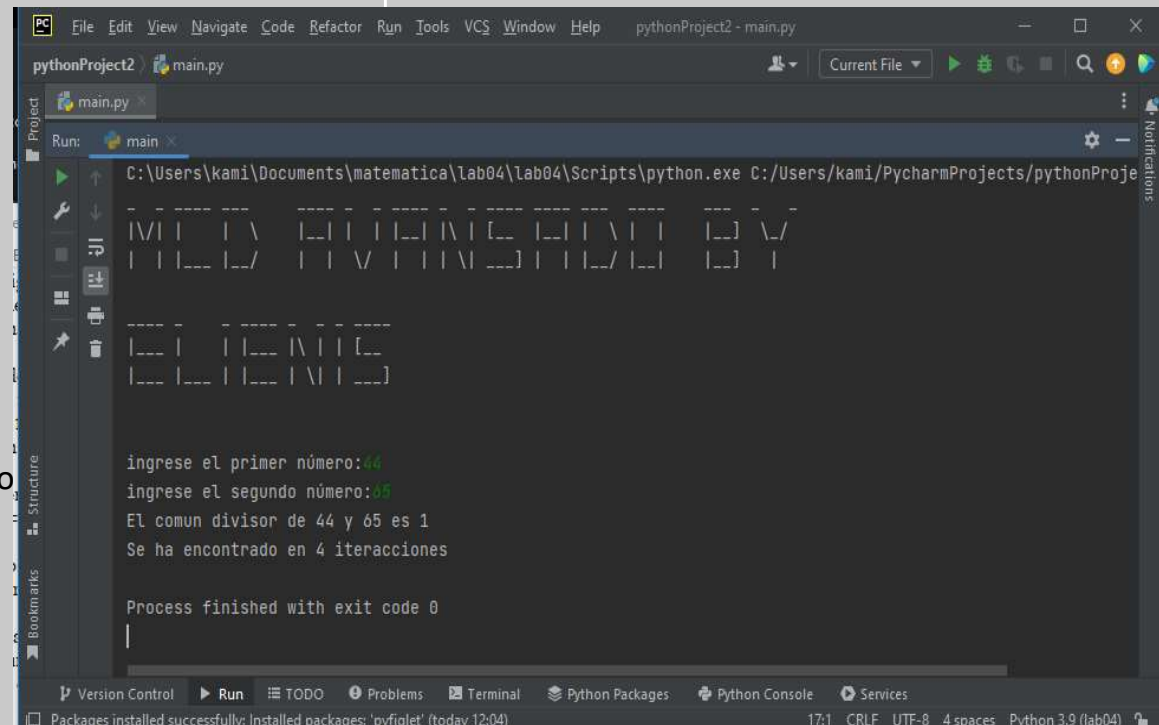
    if resto == 0:
        return (num2, iteracciones)

    # llamamos nuevamente a la función pasando como
    # segundo numero y el resto de la division
    return euclides(num2, resto, iteracciones + 1)
```

```
a = int(input("ingrese el primer número:"))
b = int(input("ingrese el segundo número:"))
num1 = a
num2 = b
```

```
comunDivisor, iteracciones = euclides(num1, num2)
```

```
print("El comun divisor de {} y {} es {}".format(num1, num2, comunDivisor))
print("Se ha encontrado en {} iteracciones".format(iteracciones))
```



```
pythonProject2 - main.py
Current File
Run: main
C:\Users\kami\Documents\matematica\lab04\lab04\Scripts\python.exe C:/Users/kami/PycharmProjects/pythonProje

MCD Euclides by Elienis

ingrese el primer número:44
ingrese el segundo número:65
El comun divisor de 44 y 65 es 1
Se ha encontrado en 4 iteracciones

Process finished with exit code 0
```

Двоичный алгоритм MCD,

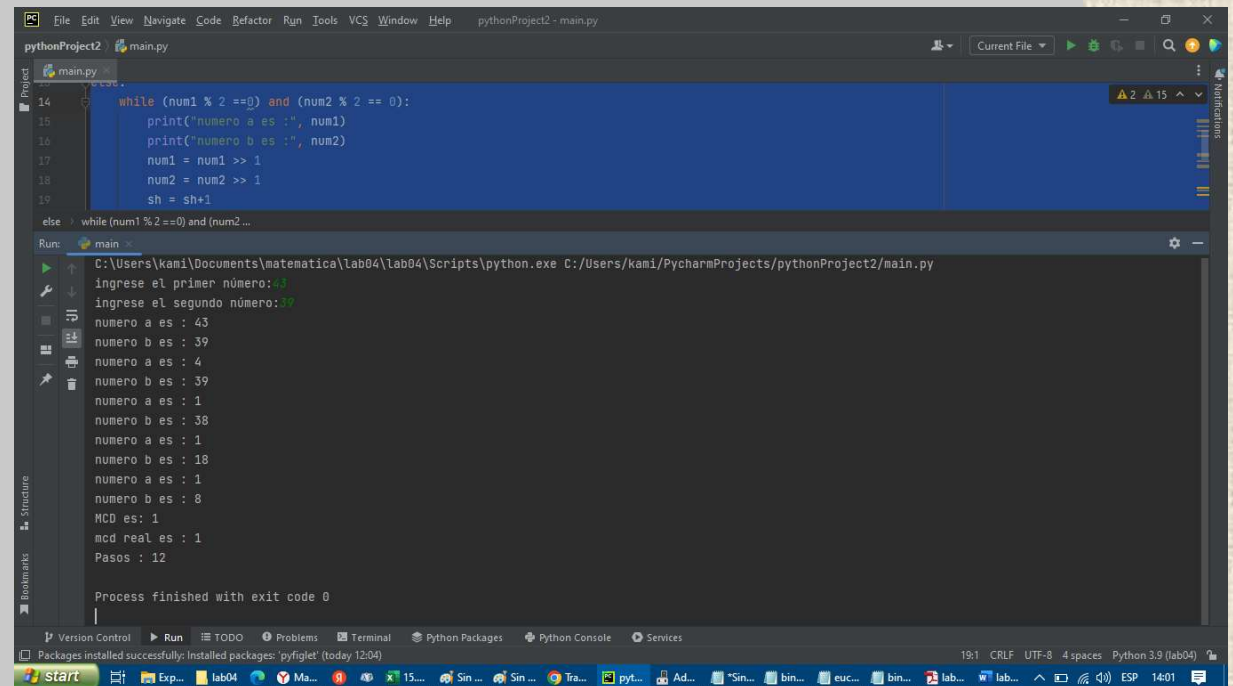
также известный как алгоритм Штейна или двоичный евклидов алгоритм, это алгоритм, который вычисляет наибольший общий делитель двух неотрицательных целых чисел. Алгоритм Штейна использует более простые арифметические операции, чем обычный алгоритм Евклида; он заменяет деление арифметическими сдвигами, сравнениями и вычитаниями.

Двоичный алгоритм MCD,

```
import math
a = int(input("ingrese el primer número:"))
b = int(input("ingrese el segundo número:"))
num1 = a
num2 = b
sh=0
steps=0

if (num1 == 0) or (num2 ==0):
    ans = 0
    #si la respuesta es 0 no necesitamos continuar
else:
    while (num1 % 2 ==0) and (num2 % 2 == 0):
        print("numero a es :", num1)
        print("numero b es :", num2)
        num1 = num1 >> 1
        num2 = num2 >> 1
        sh = sh+1
        steps = steps+1

    while (num1 != num2):
        print("numero a es :",num1)
        print("numero b es :", num2)
        while (num1 & 1 ==0):
            num1 = num1 >> 1
            steps = steps +1
        while (num2 & 1 ==0):
            num2 = num2 >> 1
            steps = steps + 1
        steps = steps + 1
    if num1 < num2:
        num2 = num2 - num1
    if num2 < num1:
        num1 = num1 - num2
    ans = num1 << sh
print("MCD es:",ans)
print("mcd real es :", math.gcd(a,b))
print("Pasos :", steps)
```



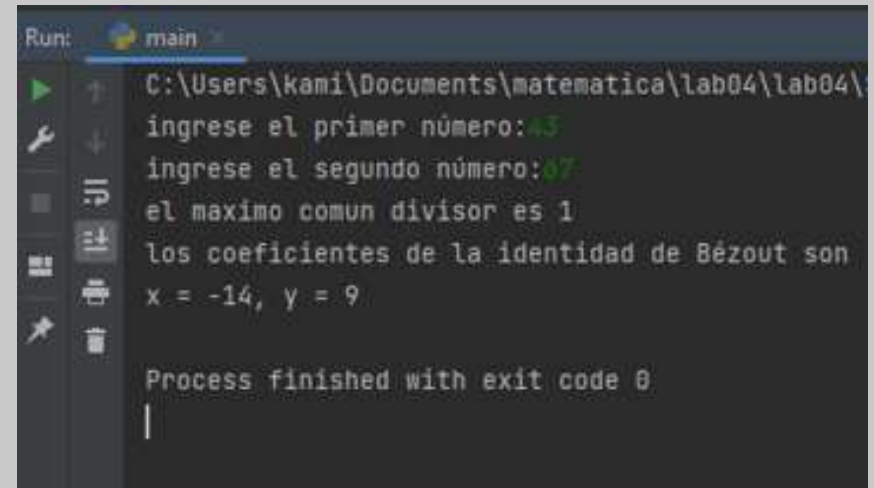
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject2 - main.py
pythonProject2 main.py
14 while (num1 % 2 ==0) and (num2 % 2 == 0):
15     print("numero a es :", num1)
16     print("numero b es :", num2)
17     num1 = num1 >> 1
18     num2 = num2 >> 1
19     sh = sh+1
else: while (num1 % 2 ==0) and (num2 ...
Run: main
C:\Users\kami\Documents\matematica\lab04\lab04\Scripts\python.exe C:/Users/kami/PycharmProjects/pythonProject2/main.py
ingrese el primer número: 43
ingrese el segundo número: 39
numero a es : 43
numero b es : 39
numero a es : 4
numero b es : 39
numero a es : 1
numero b es : 38
numero a es : 1
numero b es : 18
numero a es : 1
numero b es : 8
MCD es: 1
mcd real es : 1
Pasos : 12
Process finished with exit code 0
Version Control Run TODO Problems Terminal Python Packages Python Console Services
Packages installed successfully: Installed packages: 'pyfiglet' (today 12:04)
19:1 CRLF UTF-8 4 spaces Python 3.9 (lab04)
```


Расширенный двоичный алгоритм MCD

Алгоритм требует (n) шагов, где n -количество бит в большем из двух чисел, поскольку каждые 2 шага уменьшают хотя бы один из операндов как минимум в 2 раза. Каждый шаг включает всего несколько арифметических операций. операции (Или (1) с небольшой константой); при работе с числами размером в одно слово каждая арифметическая операция преобразуется в одну машинную операцию, поэтому количество машинных операций находится в порядке журнала

Расширенный двоичный алгоритм MCD

```
def extended_gcd(a, b):  
    if a == 0:  
        return b, 0, 1  
    else:  
        gcd, x, y = extended_gcd(b % a, a)  
        return gcd, y - (b // a) * x, x  
  
a = int(input("ingrese el primer número:"))  
b = int(input("ingrese el segundo número:"))  
x = a  
y = b  
if __name__ == '__main__':  
    gcd, x, y = extended_gcd(x, y)  
    print('el maximo comun divisor es', gcd)  
  
    print("los coeficientes de la identidad de  
Bézout son")  
    print(f'x = {x}, y = {y}')
```



```
Run: main  
C:\Users\kami\Documents\matematica\lab04\lab04\  
ingrese el primer número:43  
ingrese el segundo número:67  
el maximo comun divisor es 1  
los coeficientes de la identidad de Bézout son  
x = -14, y = 9  
  
Process finished with exit code 0
```


Microsoft Windows [Versión 10.0.19042.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\kami>

C:\Users\kami>

ВЫВОД

Хотя евклидов алгоритм используется для нахождения наибольшего общего делителя двух натуральных чисел (положительных целых чисел), его можно обобщить на действительные числа и другие математические объекты, такие как многочлены, квадратичные целые числа и Гурвица. кватернионы. В последних случаях алгоритм Евклида используется для демонстрации важнейшего свойства однозначной факторизации, а именно того, что такие числа могут быть однозначно разложены на неприводимые элементы, аналоги простых чисел. Однозначная факторизация необходима для многих тестов теории чисел.

Microsoft Windows [Versión 10.0.19042.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\kami>

C:\Users\kami>

Библиография

Perez P, B., & Acosta Velarde, R. (2020). Algoritmo Euclidiano. Ciencia Digital, 2(3), 61-74

https://hmong.es/wiki/Euclidean_algorithm

Cabrera R, Juan , Algoritmo Euclides, método de implementación (2020). Python

<https://www.techiedelight.com/es/extended-euclidean-algorithm-implementation/>

Microsoft Windows [Versión 10.0.19042.1706]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\kami>

C:\Users\kami>Большое спасибо