

Soluciones a los ejercicios de autoevaluación

The image displays two side-by-side Python IDE windows, each showing a script for an auto-evaluation exercise. The left window, titled 'Autoevaluacion 3.1.py', contains a function 'obtenermultiplos' that finds the LCM of two numbers by generating their multiples and comparing them. The right window, titled 'Python 3.5.2 Shell', shows the execution of this script with inputs 12 and 18, resulting in an LCM of 36. The left window also contains a function 'obtenerdivisores' that finds the GCD of two numbers by generating their divisors and comparing them. The right window shows the execution of this script with inputs 95 and 245, resulting in a GCD of 5 and a fraction of 19.0/49.0.

```
def obtenermultiplos (numero,elotro):
    multiplos=[]
    for i in range(elotro,0,-1):
        multiplos.append(i*numero)
    return multiplos

numero1=int(input("Introduce el primer número: "))
numero2=int(input("Introduce el segundo número: "))
multiplos1=obtenermultiplos(numero1,numero2)
multiplos2=obtenermultiplos(numero2,numero1)
for i in multiplos1:
    for z in multiplos2:
        if i==z:
            mcm=i
            encontrado=True
            break
print ("El mínimo común múltiplo es: "+str(mcm))
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Jorge/Desktop/Curso python/Autoevaluacion 3.1.py =====
Introduce el primer número: 12
Introduce el segundo número: 18
El mínimo común múltiplo es: 36
>>>
```

```
def obtenerdivisores (numero):
    divisores=[]
    for i in range(1,numero+1):
        if numero%i==0:
            divisores.append(i)
    return divisores

numero1=int(input("Introduce el numerador: "))
numero2=int(input("Introduce el denominador: "))
divisores1=obtenerdivisores(numero1)
divisores2=obtenerdivisores(numero2)
for i in divisores1:
    for z in divisores2:
        if i==z:
            mod=i
            encontrado=True
            break

fraccion={}
fraccion["Numerador"]=numero1/mod
fraccion["Denominador"]=numero2/mod
print (str (fraccion))
```

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Jorge\Desktop\Curso python\Autoevaluacion 3.2.py =====
Introduce el numerador: 95
Introduce el denominador: 245
{'Numerador': 19.0, 'Denominador': 49.0}
>>> |
```

```
Autoevaluación 3.3.py - C:\Users\Jorge\Desktop\Curso python\Autoevaluación 3.3.py (3.5.2)
File Edit Format Run Options Window Help

def obtenerdivisores (numero1,numero2):
    divisores=[]
    for i in range(1,numero1+1):
        if numero1%i==0 and numero2%i==0:
            divisores.append(i)
    return divisores

fraccion1={}
fraccion2={}
fraccion1["Numerador"]=int(input("Introduce el numerador1: "))
fraccion1["Denominador"]=int(input("Introduce el denominador1: "))
fraccion2["Numerador"]=int(input("Introduce el numerador2: "))
fraccion2["Denominador"]=int(input("Introduce el denominador2: "))
""" Se va a conseguir denominador común multiplicando el denominador
de una fracción por la otra y viceversa"""

fraccion1["Numerador"]=fraccion1["Numerador"]*fraccion2["Denominador"]
fraccion2["Numerador"]=fraccion2["Numerador"]*fraccion1["Denominador"]
fraccion1["Denominador"]=fraccion1["Denominador"]*fraccion2["Denominador"]
fraccion2["Denominador"]=fraccion2["Denominador"]*fraccion1["Denominador"]
""" Ahora dividiré por el mayor divisor común a ambas fracciones"""
print (str (fraccion1)+ "\n"+str(fraccion2))
divisores1=obtenerdivisores(fraccion1["Numerador"],fraccion1["Denominador"])
divisores2=obtenerdivisores(fraccion2["Numerador"],fraccion2["Denominador"])

for i in divisores1:
    for z in divisores2:
        if i==z:
            mcd=i

fraccion1["Numerador"]=fraccion1["Numerador"]/mcd
fraccion1["Denominador"]=fraccion1["Denominador"]/mcd
fraccion2["Numerador"]=fraccion2["Numerador"]/mcd
fraccion2["Denominador"]=fraccion2["Denominador"]/mcd
print (str (fraccion1)+ "\n"+str(fraccion2))

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 b
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Jorge\Desktop\Curso python\Autoevaluación 3.3.py
Introduce el numerador1: 12
Introduce el denominador1: 18
Introduce el numerador2: 72
Introduce el denominador2: 24
{'Numerador': 288, 'Denominador': 432}
{'Numerador': 1296, 'Denominador': 432}
{'Numerador': 2.0, 'Denominador': 3.0}
{'Numerador': 9.0, 'Denominador': 3.0}
>>> |
```

En este caso, hay un paso intermedio que se muestra. Conforme los programas se complican, es necesario decirle que saque por pantalla ciertos valores en determinados momentos para controlar que se está haciendo lo que queremos. Imagina si hubiera 2000 líneas.

Los valores correctos son los últimos.

```
Autoevaluación 3.4.py - C:\Users\Jorge\Desktop\Curso python\Autoevaluación 3.4.py (3.5.2)
File Edit Format Run Options Window Help

def mcd (numero1,numero2):
    for i in range(1,numero1+1):
        if numero1%i==0 and numero2%i==0:
            maxcomdiv=i
    return maxcomdiv

def suma (frac1,frac2,opera):
    result={}
    if opera=="+":
        result["Numerador"]=frac1["Numerador"]*frac2["Denominador"]+frac2["Numerador"]*frac1["Denominador"]
    else:
        result["Numerador"]=frac1["Numerador"]*frac2["Denominador"]-frac2["Numerador"]*frac1["Denominador"]
    result["Denominador"]=frac1["Denominador"]*frac2["Denominador"]
    return result

fraccion1={}
fraccion2={}
resultado={}
fraccion1["Numerador"]=int(input("Introduce el numerador1: "))
fraccion1["Denominador"]=int(input("Introduce el denominador1: "))
operacion=input("Introduce la operación que deseas ")
fraccion2["Numerador"]=int(input("Introduce el numerador2: "))
fraccion2["Denominador"]=int(input("Introduce el denominador2: "))
if operacion=="+" or operacion=="-":
    resultado=suma(fraccion1,fraccion2,operacion)
elif operacion=="*":
    resultado["Numerador"]=fraccion1["Numerador"]*fraccion2["Numerador"]
    resultado["Denominador"]=fraccion1["Denominador"]*fraccion2["Denominador"]
else:
    resultado["Numerador"]=fraccion1["Numerador"]*fraccion2["Denominador"]
    resultado["Denominador"]=fraccion1["Denominador"]*fraccion2["Numerador"]
    |
mcdiv=mcd(abs(resultado["Numerador"]),abs(resultado["Denominador"]))
resultado["Numerador"]=resultado["Numerador"]/mcdiv
resultado["Denominador"]=resultado["Denominador"]/mcdiv
print (str (resultado))

Python 3.5.2 Shell
File Edit Shell Debug Options Window Help

Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 b
tel)] on win32
Type "copyright", "credits" or "license()" f
>>>
===== RESTART: C:\Users\Jorge\Desktop\Curso
Introduce el numerador1: 2
Introduce el denominador1: 3
Introduce la operación que deseas +
Introduce el numerador2: 4
Introduce el denominador2: 5
{'Denominador': 15.0, 'Numerador': 22.0}
>>>
===== RESTART: C:\Users\Jorge\Desktop\Curso
Introduce el numerador1: 2
Introduce el denominador1: 7
Introduce la operación que deseas -
Introduce el numerador2: 14
Introduce el denominador2: 21
{'Denominador': 21.0, 'Numerador': -8.0}
>>>
===== RESTART: C:\Users\Jorge\Desktop\Curso
Introduce el numerador1: 3
Introduce el denominador1: 5
Introduce la operación que deseas *
Introduce el numerador2: 4
Introduce el denominador2: 10
{'Denominador': 25.0, 'Numerador': 6.0}
>>>
===== RESTART: C:\Users\Jorge\Desktop\Curso
Introduce el numerador1: 2
Introduce el denominador1: 3
Introduce la operación que deseas /
Introduce el numerador2: 1
Introduce el denominador2: 8
{'Numerador': 16.0, 'Denominador': 3.0}
>>>
```

En este caso se han optimizado ciertos algoritmos: Se ha utilizado la función **suma** para sumar y restar y se ha cambiado la función para obtener el máximo común divisor. Esto es lo que suele pasar cuando se le dan varias vueltas al mismo programa. La solución óptima no es exigible a nuestro nivel pero irás viendo que cada vez eres capaz de obtener soluciones más eficientes.

Gracias por haber seguido el curso, te deseo muchos éxitos.

Revision #1

Created 1 February 2022 11:42:18 by Equipo CATEDU

Updated 1 February 2022 11:42:19 by Equipo CATEDU