



Programación I

Programación básica con python

Profesor: Ignacio Lincolao Venegas

Ingeniería Civil Informática, UCT

Operadores aritméticos

+	Suma	$2+3$	5
-	Resta	$2-3$	-1
*	Multiplicación	$2*3$	6
**	Exponenciación	$2**3$	8
/	división	$2/3$	0.6666
//	división entera	$2//3$	0
%	módulo	$2\%3$	2

Operadores relacionales

Para los ejemplos se asumen que $a=2$, $b=10$, $c=12$

$>$	Mayor que	$a+b > c$	Falso
$<$	Menor que	$a-b < c$	Verdadero
\geq	Mayor o igual que	$a+b \geq c$	Verdadero
\leq	Menor o igual que	$c-b \leq a$	Verdadero
\neq	Diferente	$a+b \neq c$	Falso
$==$	Igual	$b==c$	Falso

Operadores lógicos

p	q	p and q	p or q	not p
Verdadero	Verdadero	Verdadero	Verdadero	Falso
Verdadero	Falso	Falso	Verdadero	Falso
Falso	Verdadero	Falso	Verdadero	Verdadero
Falso	Falso	Falso	Falso	Verdadero

Entrada y salida python

En el pseudocódigo para mostrar mensaje por pantalla existe la instrucción escribir, en el lenguaje python se utiliza la función print(), esta permite de forma general mostrar información por pantalla, mostrando texto y variables, se deben separar por comas. Ejemplo:

```
a = "tengo"  
b = 3  
print("Hola", a , b, "gatos")
```

Salida: Hola tengo 3 gatos

Existe otra forma la cual hace referencia a partir de la posición de los valores utilizando índices:

```
a = "tengo"  
b = 3  
print( "{2} gatos {1} Hola {0}".format(a,"cadena",b) )  
           0      1      2
```

Salida: Hola tengo 3 gatos

Entrada y salida python

En el pseudocódigo para recibir texto por teclado existe la instrucción leer, Para la entrada por teclado en python, existe la función `input()`, el programa al llegar a esta función se detiene y espera que se escriba algo y se pulse la tecla Intro, el valor debe ser almacenado en una variable para poder luego usarlo. Otra forma más compacta para mostrar un texto por pantalla y luego solicitar texto, es ingresando un texto a la función `input`:

```
print("Ingrese su nombre: ")
nombre = input()

nombre = input("Ingrese su nombre: ")
```

Palabras reservadas Python

En python existen un total de 28 palabras reservadas, estas palabras no pueden utilizarse para nombrar variables.

<code>and</code> <code>del</code> <code>for</code> <code>is</code> <code>raise</code> <code>assert</code> <code>if</code> <code>else</code> <code>elif</code> <code>from</code> <code>lambda</code> <code>return</code> <code>break</code> <code>global</code>	<code>not</code> <code>try</code> <code>class</code> <code>except</code> <code>or</code> <code>while</code> <code>continue</code> <code>exec</code> <code>import</code> <code>yield</code> <code>def</code> <code>finally</code> <code>in</code> <code>print</code>
---	--

Transformación de tipo de dato

Python incluye funciones nativas, que permite utilizarla sin importar o instalar paquetes adicionales, estas funciones son incorporadas dentro del mismo lenguaje, dentro de estas funciones, existen las funciones para conversión de datos.

Para convertir a cadena se utiliza la función `str()`

```
str(789) # "789"  
str(True) # "True"  
str(2+2) # "4"
```

Para convertir un entero o un string a punto flotante se utiliza la función `float()`:

```
float(2) # 2.0  
float(345) # 345.0  
float("159") # 159.0
```


Transformación de tipo de dato

Para convertir un punto flotante, un string o boolean a número entero se utiliza la función `int()`

```
int(147.021) # 147
int(True) # 1
int(False) # 0
int("16") # 16
```

Para convertir un entero, decimal o un string, a booleano se utiliza la función `bool()`, es importante destacar que `bool()` devolverá falso cuando se ingresen valores vacíos o nulos, y para cualquier otro caso equivale a true:

```
bool(0) # False
bool('') # False
bool('casa') # True
bool(2) # True
```

Redondeo de números

Una de las funciones de modo nativo en python es la función `round()` que permite redondear un número decimal a número entero, la función `round` contiene dos parámetros de entrada, el primer parámetro recibe como entrada un número flotante, y el segundo parámetro permite identificar la cantidad de dígitos después del punto respecto de la cual se debe realizar la aproximación, si este no se ingresa se asume que la función devolverá un número entero.

```
# Cuando no se ingresa el segundo parámetro
round(1.2) # 1
round(1.9) # 2
round(4.678) # 5
# cuando se ingresa una segunda el segundo parámetro de entrada
round(4.678,1) # 4.7
round(4.678,2) # 4.68
```

Math

La librería math es una librería que ofrece funciones matemáticas principalmente para el manejo de los números reales. De el conjunto de funciones existe la función `sqrt()` el cual permite resolver la raíz cuadrada de un número.

```
import math
math.sqrt(16) # 4
math.sqrt(4) # 2
```

Otra función es la función `math.pow(x,y)` lo cual devuelve el resultado de X^Y :

```
import math
math.pow(2,3) # 8
math.pow(4,3) # 64
```

También posee variables constantes:

```
math.pi # 3.14192...
math.e # 2.71828...
```

Random

La biblioteca random contiene funciones que se relacionan con los valores aleatorios. Una de las funciones es el randint() el cual devuelve un número entero entre los valores indicados, incluyéndose.

```
import random
print(random.randint(1,5)) # puede devolver 1,2,3,4,5
```

La función **random()** genera un número decimal entre 0 y 1 (puede generar 0, pero no 1)

```
import random
print(random.random()) # 0.165145432
```

La función uniform() genera un número decimal entre a y b

```
import random
print(random.uniform(6, 9)) # 8.123154169841
```

Manejo simple de string

Concatenación de cadenas

Una forma simple de concatenar cadenas de caracteres es a través del operador +, donde se pueden sumar dos o mas string para generar una nueva cadena de caracteres ejemplo:

```
fruta_1 = "Manzanas"  
fruta_2 = "Naranjas"  
frase = "Hoy compre "+ fruta_1 + "y " + fruta_2 # Hoy compre manzana y naranjas
```

Manejo simple de string

Longitud de un string

Para algunos programas se tendrá la necesidad de saber el número de caracteres que contiene un string, para esto utilizamos la función `len()` el cual devuelve el número de caracteres del string.

```
Palabra = len("Hola mundo") # 10
# Otra manera de usarlo es, es importante indicar que la salida de la función len es de
# tipo entero, por lo cual es necesario transformarlo a string para concatenarlo
print("La palabra casa contiene "+ str(len("casa"))+ " caracteres")

a = "palabra"
b = "test"
len(a)>= len(b) # devuelve True porque "palabra" contiene más caracteres que "test"
```

Manejo simple de string

Comparación de strings

La comparación de string permite saber verificar si dos string son iguales, para eso se puede utilizar el signo ==, en el caso que ambos string sean identicos eso quiere decir que las mayusculas y minusculas si pueden afectar, por ejemplo “Casa” y “casa” son distintas y devolverá false:

```
clave1 = "clave123"  
clave2 = "clave567"  
clave3 = "Clave123"  
clave4 = "clave123"  
  
clave1 == clave4 # retorna True  
clave1 == clave2 # retorna false  
clave1 == clave3 # retorna false
```

Manejo simple de string

Conversión de palabras a mayúsculas y minúsculas

En algunas ocasiones es necesario cambiar los caracteres de un string a minúsculas o mayúsculas para eso existen 2 funciones `lower` que convierte a minúscula y `upper` que convierte a mayúscula:

```
palabra_mayuscula = upper("hola mundo") # HOLA MUNDO
palabra_minuscula = lower("HOLA MUNDO") # hola mundo
palabra_minus = lower("HOLA mundo") # hola mundo
palabra_mayus = upper("HOLA mundo") # HOLA MUNDO
```


Manejo simple de string

Extracción un substring de un string

Se puede extraer una porción de un string a través del uso de posiciones del string, el primer carácter tiene la posición 0 y luego se va aumentando 1 en 1.

```
cadena = "Tareas diarias"
# T a r e a s   d i a r i a s
# 0 1 2 3 4 5 6 7 8 9 10 11 12 13

print(cadena[7:]) # diarias
print(cadena[1:]) # areas diarias
print(cadena[:6]) # Tareas
print(cadena[7:10]) # dia
```

Uso del if

El uso de la sentencia if permite ejecutar un bloque de código si, y sólo si, se cumple la condición se evalúa como True, la estructura de la sentencia if es:

```
if condición:  
    bloque de código
```

La condición se evalúa con el uso de operadores condicionales y operadores lógicos.

```
a = 2  
b = 3  
if a < b:  
    print("a es menor que b")
```

Sentencia if else

En ocasiones es necesario ejecutar un bloque de código si cumple la condición y ejecutar otro en caso que no cumpla la condición para esto se utiliza la sentencia else:

```
a = 2
b = 3
if a < b:
    print("a es menor que b")
else:
    print("b es mayor que a")
```

Sentencia if elif else

En el caso que exista una situación que dependa de múltiples condiciones se pueden utilizar una sentencia de if compuesta:

```
a = 2
b = 3
if a < b:
    print("a es menor que b")
elif a > b:
    print("a es mayor a b")
else:
    print("a y b son iguales")
```

Sentencia if anidadas

La sentencia if también puede ser anida dentro de otro if:

```
a = 2
b = 3
if a < b:
    print("a es menor que b")
else:
    if a > b:
        print("a es mayor a b")
    else:
        print("a y b son iguales")
```

Sentencia if

La sentencia if también puede estar compuesta por múltiples operadores lógicos:

```
a = 2
b = 3
if a < b and b < 10:
    print("a es menor que b y menor que 10")
elif b > a and b >= 10:
    print("b es mayor que a y mayor que 10")
else:
    print("no cumple ninguna de las condiciones")

if a < b or b >= 3:
    print("b cumple 1 de las dos condiciones")
```

Problemas encontrados

El análisis y pseudocódigo son diferentes, en el análisis se detalla las variables de entrada, salida y los procesos en cambio en el pseudocódigo se diseña la solución generando un algoritmo que resuelva el problema. Ejemplo: Dada una base y altura de un rectángulo calcular el área.

Análisis	Diseño (pseudocódigo)
<p>Entrada: base, altura</p> <p>Salida: area</p> <p>Operaciones: area = base*altura</p>	<pre>Algoritmo calcular_area leer base, altura area <- base * altura escribir area finalgoritmo</pre>

Problemas encontrados

En las variables de entrada y salida no pueden ir operaciones aritméticas solo el nombre de la variable, las operaciones aritméticas solo se deben hacer en la sección de operaciones. Además los nombres de las variables no deben llevar espacio, en el caso de ser dos variables deben separarse por “,” y no usando el “y”:

Incorrecto	Correcto
<p>Entrada: numero_a + numero_b</p> <p>Entrada: altura y base</p> <p>Entrada: hipotenusa elevada</p> <p>salida: numero_a + numero_b</p> <p>leer numero_a+numero_b</p> <p>leer cateto**2</p> <p>escribir numero_a+numero_b</p>	<p>Entrada: numero_a, numero_b</p> <p>Entrada: altura, base</p> <p>Entrada: hipotenusa_elevada</p> <p>Salida: suma_ab</p> <p>leer numero_a, numero_b</p> <p>leer cateto</p> <p>escribir suma_ab</p>

Problemas encontrados

Las operaciones no se pueden estar al lado izquierdo de la operación, solo debe estar el nombre de la variable donde se almacenará el resultado. En las operaciones debe existir al menos una variable que se guarde el resultado

Incorrecto	Correcto
<code>numero_a + numero_b= suma_ab</code>	<code>suma_ab = numero_a + numero_b</code>
<code>numero_a - numero_b= suma_ab</code>	<code>suma_ab = numero_a - numero_b</code>
<code>numero_a / numero_b= suma_ab</code>	<code>suma_ab = numero_a / numero_b</code>
<code>numero_a * numero_b= suma_ab</code>	<code>suma_ab = numero_a * numero_b</code>
<code>numero_a + numero_b</code>	

Problemas encontrados

El uso de \geq y \leq es para operaciones lógicas, se debe utilizar en las bifurcaciones o ciclos, como por ejemplo “si $a \geq b$ entonces”:

Incorrecto	Correcto
<pre>Algoritmo distancia leer numero_1>=numero2 leer numero_2<=numero_1 distancia=numero_1-numero_2 escribir distancia finalgoritmo</pre>	<pre>Algoritmo distancia leer numero_1, numero_2 si numero_1 >= numero_2 entonces distancia <- numero_1 - numero_2 sino distancia <- numero_2 - numero_1 escribir distancia finalgoritmo</pre>

Problemas encontrados

El algoritmo del pseudocódigo debe ir dentro de bloque el cual está definido por la palabra “**Algoritmo**” continuado por el nombre del algoritmo, y para cerrarlo se debe escribir **finalgoritmo**:

Incorrecto	Correcto
<pre>leer base, altura area <- base * altura escribir base</pre>	<pre>Algoritmo calcular_area leer base, altura area <- base * altura escribir area finalgoritmo</pre>

Pregunta 1

Problema: Genere un programa que recibe una cantidad de minutos, y me muestra la cantidad de horas y minutos que corresponde.

Análisis	Diseño (pseudocódigo)
<p>Entrada:Minutos</p> <p>Salida:cantidad_horas, minutos_restantes</p> <p>Operaciones:</p> <p>Horas = minutos // 60</p> <p>minutos_restantes = minutos % 60</p>	<pre>Algoritmo Horas_minutos leer minutos horas <- trunc(minutos / 60) minutos <- minutos mod 60 escribir horas, minutos finalgoritmo</pre>

Pregunta 2

Problema: Dados los catetos de un triángulo rectángulo, calcular su hipotenusa.

Análisis	Pseudocódigo
<p>Entrada:cateto_1, cateto_2</p> <p>Salida:hipotenusa</p> <p>Operaciones:</p> <p>$\text{hipotenusa_elev} = \text{cateto_1}^2 + \text{cateto_2}^2$</p> <p>$\text{hipotenusa} = \text{hipotenusa_elev}^{(1/2)}$</p>	<pre>Algoritmo hipotenusa leer cateto_1, cateto_2 hipotenusa_elev <- cateto_1**2 + cateto_2**2 hipotenusa <- hipotenusa_elev**(1/2) escribir hipotenusa finalgoritmo</pre>

Pregunta 3

Problema: Dados dos números, mostrar la suma, resta, división y multiplicación de ambos.

Análisis	Pseudocódigo
<p>Entrada:numero_1, numero_2</p> <p>Salida: suma, resta, division, multiplicacion</p> <p>Operaciones:</p> <p>suma = numero_1 + numero_2</p> <p>resta = numero_1 - numero_2</p> <p>division = numero_1 / numero_2</p> <p>multiplicacion = numero_1 * numero_2</p>	<p>Algoritmo operaciones</p> <p> leer numero_1, numero_2</p> <p> suma <- numero_1 + numero_2</p> <p> resta <- numero_1 - numero_2</p> <p> division <- numero_1 / numero_2</p> <p> multiplicacion <- numero_1 * numero_2</p> <p> escribir suma, resta, division,multiplicacion</p> <p>finalgoritmo</p>

Pregunta 4

Problema: Se reciben 2 números distintos, se debe mostrar la distancia entre ellos.

Análisis	Pseudocódigo
<p>Entrada:numero_1, numero_2</p> <p>Salida:distancia</p> <p>Operaciones:</p> <p>$\text{distancia} = \text{numero}_1 - \text{numero}_2$</p> <p>$\text{distancia} = \text{numero}_2 - \text{numero}_1$</p>	<pre>Algoritmo distancia_numeros leer numero_mayor, numero_menor si numero_1 > numero_2 entonces distancia = numero_1 - numero_2 sino distancia = numero_2 - numero_1 escribir distancia finalgoritmo</pre>

Pregunta 5

Problema: Calcular el perímetro y área de un rectángulo dada su base y su altura.

Análisis	Pseudocódigo
<p>Entrada: base, altura</p> <p>Salida: perimetro, area</p> <p>Operaciones:</p> $\text{perimetro} = (\text{base} + \text{altura}) * 2$ $\text{area} = \text{base} * \text{altura}$	<pre>Algoritmo base_altura leer base, altura perimetro <- (base + altura) * 2 area <- base * altura escribir perimetro, area finalgoritmo</pre>

Problema 1

```
1  """
2  Problema: Genere un programa que recibe una cantidad de minutos, y me muestra la cantidad de horas y minutos que corresponde.
3
4  Analisis
5  Entrada:base, altura
6
7  Salida: area
8
9  Operaciones:
10 area = base*altura
11
12 Diseño
13 Algoritmo calcular_area
14     leer base, altura
15     area <- base * altura
16     escribir area
17 finalgoritmo
18
19 """
20 # Programa
21 minutos = int(input("Ingrese la cantidad de minutos:"))
22 horas = minutos // 60
23 minutos = minutos % 60
24 print("La cantidad de horas son",horas,"y sus minutos restantes son:",minutos)
25
```

Problema 2

```
test.py > ...
1  """
2  Problema Dados los catetos de un triángulo rectángulo, calcular su hipotenusa.
3
4  Analisis
5  Entrada:cateto_1, cateto_2
6
7  Salida:hipotenusa
8
9  Operaciones:
10 hipotenusa_elev = cateto_1**2+cateto_2**2
11 hipotenusa = hipotenusa_elev**(1/2)
12
13 Diseño
14  Algoritmo hipotenusa
15      leer cateto_1, cateto_2
16      hipotenusa_elev <- cateto_1**2 + cateto_2**2
17      hipotenusa <- hipotenusa_elev**(1/2)
18      escribir hipotenusa
19  finalgoritmo
20
21
22 """
23 # Programa
24 import math
25 cateto_1 = float(input("Ingrese el primero cateto:"))
26 cateto_2 = float(input("Ingrese el segundo cateto:"))
27 hipotenusa = math.sqrt(cateto_1**2+cateto_2**2)
28 print("El valor de la hipotenusa es:", hipotenusa)
29 |
```

Problema 3

```

** **
Problema: Dados dos números, mostrar la suma, resta, división y multiplicación de ambos.

Análisis
Entrada: numero_1, numero_2

Salida: suma, resta, division, multiplicacion
Operaciones:
suma = numero_1 + numero_2
resta = numero_1 - numero_2
division = numero_1 / numero_2
multiplicacion = numero_1 * numero_2

Diseño

Algoritmo operaciones
    leer numero_1, numero_2
    suma <- numero_1 + numero_2
    resta <- numero_1 - numero_2
    division <- numero_1 / numero_2
    multiplicacion <- numero_1 * numero_2
    escribir suma, resta, division, multiplicacion
finalgoritmo
** **

# Programa
numero_1 = float(input("Ingrese el primero numero:"))
numero_2 = float(input("Ingrese el segundo numero:"))
suma = numero_1 + numero_2
resta = numero_1 - numero_2
division = numero_1 / numero_2
multiplicacion = numero_1 * numero_2
print("La suma es:", suma, "su resta es:", resta, "su division es:", division, "su multiplicacion es:", multiplicacion)

```

Problema 4

```
"""
Problema: Se reciben 2 números distintos, se debe mostrar la distancia entre ellos.
```

```
Analisis
```

```
Entrada:numero_mayor, numero_menor
```

```
Salida:distancia
```

```
Operaciones:
```

```
distancia = numero_mayor - numero_menor
```

```
Diseño
```

```
Algoritmo distancia_numeros
```

```
    leer numero_mayor, numero_menor
```

```
    si numero_mayor < numero_menor entonces
```

```
        distancia = numero_menor - numero_mayor
```

```
    sino
```

```
        distancia = numero_mayor - numero_menor
```

```
    escribir distancia
```

```
finalgoritmo
```

```
"""
```

```
# Programa
```

```
numero_1 = int(input("Ingrese el primero numero:"))
```

```
numero_2 = int(input("Ingrese el segundo numero:"))
```

```
if numero_1 < numero_2:
```

```
    distancia = numero_2 - numero_1
```

```
else:
```

```
    distancia = numero_1 - numero_2
```

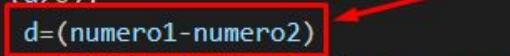
```
print("la distancia entre los 2 numeros es:", distancia)
```

Problema 5

```
1  """
2  Problema: Calcular el perímetro y área de un rectángulo dada su base y su altura.
3
4  Analisis
5  Entrada:base, altura
6
7  Salida: perimetro, area
8
9  Operaciones:
10 perimetro = (base + altura) * 2
11 area = base * altura
12
13 Diseño
14 ▾ Algoritmo base_altura
15     leer base, altura
16     perimetro <- (base + altura) * 2
17     area <- base * altura
18     escribir perimetro, area
19 finalgoritmo
20
21
22 """
23 # Programa
24 base = float(input("Ingrese la base:"))
25 altura = float(input("Ingrese la altura:"))
26 perimetro = (base+altura)*2
27 area = base*altura
28 print("la base es", base,"y la altura es", altura)
29
```

Recomendaciones: Reutilizar variables

```
4  numero2=int(input())
5  d=(numero1-numero2)
6  ✓ if (d>0):
7      d=(numero1-numero2)
8      print("La distancia de los numeros es ", d)
9  ✓ else:
10     d=d*(-1)
11     print ("La distancia de los numeros es ", d)
12
```



```
14  d=(numero1-numero2)
15  ✓ if (d>0):
16      print("La distancia de los numeros es ", d)
17  ✓ else:
18      d=d*(-1)
19      print ("La distancia de los numeros es ", d)
20
21
```

Recomendaciones: Importaciones al inicio

```
8  print("Ingrese la medida del primer cateto")
9  c1=int(input())
10 print("Ingrese la medida del segundo cateto")
11 c2=int(input())
12 import math
13 hip=(math.sqrt ((c1**2) + (c2**2)))
14 print("La medida de la hipotenusa de su triangulo es ")
15 print(hip)
16
```

```
9  import math
10 print("Ingrese la medida del primer cateto")
11 c1=int(input())
12 print("Ingrese la medida del segundo cateto")
13 c2=int(input())
14 hip=(math.sqrt ((c1**2) + (c2**2)))
15 print("La medida de la hipotenusa de su triangulo es ")
16 print(hip)
17
```

Recomendaciones: Solo importar cuando se usa la librería

```
1 import math
2 print("Digite la base de su rectángulo")
3 base=int(input())
4 print("Digite la períméto de su rectángulo")
5 alt=int(input())
6 perimetro= ((base+alt)*2)
7 area= (base*alt)
8 print("El perímetro de su rectángulo es: ",perimetro)
9 print("El área de su rectángulo es: ", area)
```

```
1 print("Digite la base de su rectángulo")
2 base=int(input())
3 print("Digite la períméto de su rectángulo")
4 alt=int(input())
5 perimetro= ((base+alt)*2)
6 area= (base*alt)
7 print("El perímetro de su rectángulo es: ",perimetro)
8 print("El área de su rectángulo es: ", area)
9
```


Recomendaciones: Establecer el tipo de dato en la entrada

```
5 print ("Ingresa el primer cateto")
6 cat1 = input()
7 print ("Ingresa el segundo cateto")
8 cat2 = input()
9
10 cat1r = int(cat1)*int(cat1)
11 cat2r = int(cat2)*int(cat2)
12 hip = int(cat1r)+int(cat2r)
13
14 hipr = int((hip)**0.5)
15 print ("La hipotenusa es", hipr)
```

```
5 print ("Ingresa el primer cateto")
6 cat1 = int(input()) # Tambien puede ser float
7 print ("Ingresa el segundo cateto")
8 cat2 = int(input())
9
10 cat1r = cat1 * cat1
11 cat2r = cat2 * cat2
12 hip = cat1r+cat2r
13
14 hipr = hip**0.5
15 print ("La hipotenusa es", hipr)
```

Recomendaciones: Si no entiendes para qué es la librería no importes

```
1 import math
2 from re import A
3
4
5 A = int(input("ingrese valor de cateto 1 "))
6 B = int(input("ingrese valor de cateto 2 "))
7
8 hipotenusa = (A**2) + (B**2)
9 Y = math.sqrt(hipotenusa)
10 print("el valor de la hipotenusa es de", Y)
```

```
1 # -*- coding: utf-8 -*-
2 from tokenize import Double
3
4
5 print("Inserte 1er numero: ")
6 n1=float(input())
7 print("Inserte 2do numero: ")
8 n2=float(input())
9 if n1==n2:
10     print ("Usted a ingresado dos numeros iguales")
11 else:
12     d= n1-n2
13     if d>=0:
14         print ("La distancia es:",d)
15     else:
16         d=d*(-1)
17     print ("la distancia es:",d)
```

Recomendaciones: Ojo con las tabulaciones

```
1 num1 = input ("Ingrese un numero ")
2     new_num1 = int(num1)
3 num2 = input ("Ingrese un segundo numero ")
4     new_num2 = int(num2)
5
6 print ("Suma")
7 print (new_num1 + new_num2)
8 print ("Resta")
9 print (new_num1 - new_num2)
10 print ("Multiplicacion")
11 print (new_num1 * new_num2)
12 print ("division")
13 print (new_num1 // new_num2)
```

```
1 num1 = input ("Ingrese un numero ")
2 new_num1 = int(num1)
3 num2 = input ("Ingrese un segundo numero ")
4 new_num2 = int(num2)
5
6 print ("Suma")
7 print (new_num1 + new_num2)
8 print ("Resta")
9 print (new_num1 - new_num2)
10 print ("Multiplicacion")
11 print (new_num1 * new_num2)
12 print ("division")
13 print (new_num1 // new_num2)
```