Trabajo Práctico 4 - Modularización

Funciones

1. Crear una función que tome un argumento numérico y devuelva ese número elevado al cuadrado. Luego de haber creado la función, pedirle al usuario 5 números, de a uno, e ir mostrando cada número elevado al cuadrado (utilizando dicha función).

#Ejercicio 1

import math

a=0

b=0

def potencia2(a,b):

b=a\*\*2

return b

for i in range (0,5):

a=int(input("Por favor, Ingrese un número "))

b=potencia2(a, b)

print (a,"^ 2 =",b)

1. Crear una función llamada *es\_positivo*que tome un número como argumento y devuelva verdadero o falso, como valores lógicos, si el número es positivo o no.

#Ejercicio 2

a=0

b=False

def es\_positivo(a,b):

if (a>0):

return True

if (a<0):

return False

a=int(input("Por favor, Ingrese un número "))

c=es\_positivo(a, b)

if (c==True):

print (a, "es un número positivo")

if (c==False):

print (a, "es un número negativo")

1. Crear una función llamada *iguales,* que tome dos palabras como parámetros, y determine si son iguales o no. Devolviendo verdadero *(true)*si lo son, o falso *(false)*en caso contrario.

#Ejercicio 3

a=""

b=""

c=False

def iguales(a,b):

if (a==b):

return True

if (a!=0):

return False

a=str(input("Por favor, Ingrese una palabra "))

b=str(input("Por favor, Ingrese otra palabra "))

c=iguales(a, b)

if (c==True):

print ("Las palabras ingresadas son iguales")

if (c==False):

print ("Las palabras ingresadas son distintas")

1. Crear una función llamada *signo,* que tome un número y devuelva 1 si este es positivo y -1 si este es negativo.

#Ejercicio 4

a=0

b=False

def signo(a,b):

if (a>0):

return 1

if (a<0):

return -1

a=int(input("Por favor, Ingrese un número "))

c=signo(a, b)

if (c==1):

print (a, "es un número positivo")

if (c==-1):

print (a, "es un número negativo")

1. Crear una función llamada escalón, que tome un número y devuelva 1 si este es positivo y 0 si este es negativo.

#Ejercicio 5

a=0

b=False

def escalon(a,b):

if (a>0):

return 1

if (a<0):

return 0

a=int(input("Por favor, Ingrese un número "))

c=escalon(a, b)

if (c==1):

print (a, "es un número positivo")

if (c==0):

print (a, "es un número negativo")

1. Crear una función llamada delta\_de\_dirac que tome dos números enteros y devuelva 1 si ambos números son iguales, y 0 sino.

#Ejercicio 6

a=0

b=0

def delta\_de\_dirac(a,b):

if (a==b):

return 1

if (a!=b):

return 0

a=int(input("Por favor, Ingrese un número "))

b=int(input("Por favor, otro número "))

c=delta\_de\_dirac(a, b)

if (c==1):

print ("Los numeros ingresados son Iguales")

if (c==0):

print ("Los números ingresados son Distintos")

1. Crear una función llamada raíz\_uno, que tome tres parámetros: a, b, c. Y calcule solo la primera raíz de la función cuadrática. ¿La función debería devolver un valor numérica entero o con decimales?

#Ejercicio 7

from math import sqrt

d=float()

def raiz\_uno(a, b, c, d):

discriminante = b \*\* 2 - 4 \* a \* c

if discriminante >= 0:

raiz = sqrt(discriminante)

d=((-b + raiz) / (2 \* a))

return d

a=float(input("Por favor, Ingrese el término a "))

b=float(input("Por favor, Ingrese el término b "))

c=float(input("Por favor, Ingrese el término c "))

d=raiz\_uno(a, b, c, d)

print ("La primera raiz de",a,"x^2 +",b,"x + ",c,"= 0 es x =",d)

1. Crear una función que tome tres números como parámetros *n, a, b,* y devuelva verdadero o falso, según *n*pertenece o no al intervalo cerrado [a, b]

#Ejercicio 8

a=0

b=0

def pertenece(a,b):

if (n>a) and (n<b):

print ("El Número Ingresado pertenece al intervalo [",a,";",b,"]")

return True

else:

print ("El Número Ingresado no pertenece al intervalo [",a,";",b,"]")

return False

a=int(input("Por favor, Ingrese el límite inferior del intervalo "))

b=int(input("Por favor, Ingrese el límite superior del intervalo "))

while (b<a):

b=int(input("Por favor, Ingrese el límite superior del intervalo "))

n=int(input("Por favor, Ingrese un número "))

c=pertenece(a, b)

1. Crear una función que tome una palabra y devuelva la cantidad de vocales que tiene. Por ejemplo, si se le da el siguiente argumento a la función: ’hola’ la función debería devolver 2.

#Ejercicio 9

b=int()

c=0

d=0

i=0

def vocales(a,b):

for i in a:

if i in "aeiouAEIOU":

b=b+1

return b

a=str(input("Por favor, Ingrese una palabra "))

b=vocales(a, b)

print ("En la palabra ingresada hay ",b,"vocales")

1. Crear una función que convierta una temperatura en Fahrenheit, en su temperatura equi­valente, en grados Celsius.

Recordar que la relación entre ambas cantidades es: Tc = (5)(Tf — 32)

Pedirle luego, al usuario temperaturas en Fahrenheit, unas 10 e ir mostrándole su conversión a grados centígrados.

#Ejercicio 10

a=0

b=0

f=int()

def temperatura(a,b):

c=(f-32)\*5/9

return c

f=int(input("Por favor, Ingrese la temperatura en Fahrenheit "))

c=temperatura (a, b)

round(c)

print ("La temperatura en grados Celsius es:",c)

1. Crear una función que tome dos palabras como parámetros, y devuelva el texto resultante de concatenar ambas palabras.

#Ejercicio 11

a=""

b=""

c=""

def concatenar(a,b):

c= a + " " + b

return c

a=str(input("Por favor, Ingrese una palabra "))

b=str(input("Por favor, Ingrese otra palabra "))

c=concatenar(a, b)

print ("El reultado de concatenar las palabras ingresadas es:",c)

1. Crear ahora una segunda función, que tome un tercer argumento extra, y haga lo mismo que la función del punto anterior, pero esta vez, utilizando el tercer argumento para saber si debe agregar o no un espacio entre medio de las dos palabras a concatenar. ¿Qué tipo de dato utilizaría para ese tercer argumento?
2. Crear una función que tome como argumentos una frase y una letra, y determine cuantas veces esta esa letra en dicha frase.

Ejemplo: Si se le pasan los siguientes dos argumentos ’Joker’ y ’k’ la función debería devolver el valor numérico 1, si la función en cambio, recibe los valores ’Pedrito’ y ’a’ debería devolver 0, y asó suc.

#Ejercicio 13

b=int()

def buscar\_letra (b):

a=len(frase)

frase==frase.lower()

letra==letra.lower()

i=0

for i in range (0,a):

if (frase[i]==letra):

b=b+1

return b

frase=str(input("Por favor, Ingrese una frase "))

letra=str(input("Por favor, Ingrese una letra "))

b=buscar\_letra(b)

print ("En la frase ingresada hay",b,"letras",letra)

1. Crear una funcion llamada capitalizar que tome una palabra como argumento y devuelva una palabra con la primer letra en mayusculas, y resto de las letras en mimísculas, de la palabra original.
2. Crear una función que tome una lista de valores numéricos como argumento, de dos elementos nada más, y devuelva la lista ordenada. En el caso de Python, ¿Necesitó utilizar una segunda lista, auxiliar para modificarla, o pudo devolver la lista original, el argumento que recibió, modificado y ordenado?

#Ejercicio 15

a=0

b=0

def ordenar(a,b):

numeros.sort ()

return b

num=int(input("Por favor, Ingrese un número de dos dígito "))

while ((num<10) or (num>99)):

num=int(input("Por favor, Ingrese un número de dos dígitos "))

numeros = ['cero','uno','dos','tres','cuatro','cinco','seis','siete','ocho','nueve']

print ("El número ingresado es el ",numeros [num])

1. Realizar una función que tome dos números: a, b y devuelva la cantidad de números pares que hay en el intervalo cerrado [a, b]. Controlar que a <= b

#Ejercicio 16

import math

a=0

b=0

c=0

def pares(a,b):

d=0

for a in range (a,b):

c=a%2

if (c==0):

d=d+1

return d

a=int(input("Por favor, Ingrese el límite inferior del intervalo "))

b=int(input("Por favor, Ingrese el límite superior del intervalo "))

while (b<=a):

b=int(input("Por favor, Ingrese el límite superior del intervalo "))

d=pares(a, b)

print ("La cantidad de números pares en el intervalo [",a,";",b,"] son:",d)