TC1028
Pensamiento computacional para ingeniería

Darío Cuauhtémoc Peña Mariano

Tarea 6 programa 1

Funcionamiento:

Este programa imprime los datos de una lista, dentro de la lista tenemos, listas, diccionarios y tuplas, con estos datos vamos a crear una tupla y sumar 3 numeros del arreglo.

Pseudocódigo:

Variables:

```
x<--  [True, False, True,("1","2","3",{"4":[5,6,7,8]})] ,9 ,10 , {"11":[12,[13,[14,15]]]} ]
n12<--x[3]["11"][0]
n13<-- x[3]["11"][1][0]
n14 <--x[3]["11"][1][1][0]
n15 <--x[3]["11"][1][1][1]
n1 <--x[0][3][0]
n2 <--x[0][3][1]
n3 <--x[0][3][2]
n5 <--x[0][3][3]["4"][0]
n6 <--x[0][3][3]["4"][1]
n7 <--x[0][3][3]["4"][2]
n8 <--x[0][3][3]["4"][3]
Inicio

tuplanumeros<--(n1,n2,n3)
suma <-- n5+n6+n7+n8

mostrar("numero 12:", n12)
mostrar("numero 13:",n13)
mostrar("numero 14:",n14)
mostrar("numero 15:",n15)
mostrar("numero 1:",n1)
mostrar("numero 2:",n2)
mostrar("numero 3:",n3)
mostrar("tupla", tuplanumeros)
mostrar("numero 5:",n5)
mostrar("numero 6:",n6)
mostrar("numero 7:",n7)
mostrar("numero 8:",n8)
mostrar("suma de numeros:",suma)
regresar 0
Fin
```

Funcionamiento del programa:

python > tareas > tarea6 > programa1.py > ...

```python
x=[ [True, False, True,("1","2","3",{"4":[5,6,7,8]})] ,9 ,10 , {"11":[12,[13,[14,1
#tupla ()
#lista []
#diccionario{}

n12= x[3]["11"][0]
n13= x[3]["11"][1][0]
n14= x[3]["11"][1][1][0]
n15 =x[3]["11"][1][1][1]

n1 =int(x[0][3][0])
n2=int(x[0][3][1])
n3 =int(x[0][3][2])

tuplanumeros= (n1,n2,n3)

n5 =x[0][3][3]["4"][0]
n6=x[0][3][3]["4"][1]
n7 =x[0][3][3]["4"][2]
n8 =x[0][3][3]["4"][3]

suma = n5+n6+n7+n8

print("numero 12:", n12)
print("numero 13:",n13)
print("numero 14:",n14)
print("numero 15:",n15)
print("numero 1:",n1)
print("numero 2:",n2)
print("numero 3:",n3)
print("tupla", tuplanumeros)
print("numero 5:",n5)
print("numero 6:",n6)
print("numero 7:",n7)
print("numero 8:",n8)
print("suma de numeros:",suma)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE COMMENTS Python

```
PS D:\tec\trabajosProg> & C:/Users/dari-/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/tec/t
josProg/python/tareas/tarea6/programa1.py
numero 12: 12
numero 13: 13
numero 14: 14
numero 15: 15
numero 1: 1
numero 2: 2
numero 3: 3
tupla (1, 2, 3)
numero 5: 5
numero 6: 6
numero 7: 7
numero 8: 8
suma de numeros: 26
PS D:\tec\trabajosProg>
```

Darío Cuauhtémoc Peña Mariano
Tarea 6 programa 2

Funcionamiento:

El programa usa 3 funciones, la primera “crearMatriz” recibe como argumento dos enteros a,b para crear una matriz con numeros aleatorios del 1 al 5, donde su longitud será del tamaño axb, la segunda función “impresionMatriz” recibe como argumento una matriz y la imprime por pantalla, la ultima función “sumaTrianguloSuperior” recibe como argumento una matriz, calcula su triangular superior, suma los elementos y regresa el resultado de la suma, al iniciar el programa pediremos al usuario que ingrese la longitud de la matriz que se creará con nuestra función, con esa matriz creada usaremos nuestra función de imprimir para imprimirla y también usaremos nuestra función para calcular la triangular superior terminando con mostrar el resultado por terminal.

Pseudocódigo:

Paquetes: “random”

|
Inicio

```
funcion crearMatriz(a,b):
  matriz <-- []
  for x in range(a):
    listaLit <-- []
    for y in range(b):
      dato <-- random.randint(1,5)
      listaLit.agregar(dato)
    matriz.agregar(listaLit)
  mostrar(" ")
  regresarmatriz

funcion impresionMatriz(matriz):
  for x in range(len(matriz)):

    for y in range(len(matriz[0])):
     mostrar(matriz[x][y], end= " ")
    mostrar("")
  mostrar(" ")


funcion sumaTrianguloSuperior(matriz):
  suma<--0
  for x in range(longitud(matriz)):
  for y in range(llongitud(matriz[0])):

    if x < y:

      suma <-- suma + matriz[x][y]

  regresar suma

elemtosMatriz<--int(input("ingresa la longitud de la matriz"))
matriz <--crearMatriz(elemtosMatriz, elemtosMatriz)
impresionMatriz(matriz)
resultado<--sumaTrianguloSuperior(matriz)
mostrar ("suma de triangular de la matriz:", resultado)

regresar 0
```

Fin

Funcionamiento del programa:

python > tareas > tarea6 > programa2.py > ...

```python
import random

def crearMatriz(a,b):
    matriz = []
    for x in range(a):
        listaLit = []
        for y in range(b):
            dato = random.randint(1,5)
            listaLit.append(dato)
        matriz.append(listaLit)
    print(" ")
    return matriz


#impresion matriz
def impresionMatriz(matriz):
    for x in range(len(matriz)):

        for y in range(len(matriz[0])):
          print(matriz[x][y], end= "  ")
        print("")
    print(" ")


def sumaTrianguloSuperior(matriz):
    suma=0
    for x in range(len(matriz)):
     for y in range(len(matriz[0])):

        if x < y:
            # x < y triangular superior
            #x>y trinagular inferior
            #x==y diagonal principal
            suma = suma + matriz[x][y]

    return suma

elemtosMatriz= int(input("ingresa la longitud de la matriz"))
matriz =crearMatriz(elemtosMatriz, elemtosMatriz)
impresionMatriz(matriz)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE COMMENTS Python

```
PS D:\tec\trabajosProg> & C:/Users/dari-/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/tec/trabajosProg/python/tareas/tarea6/programa2.py
ingresa la longitud de la matriz4

5  3  2  3
1  1  4  4
3  2  1  1
4  1  5  1

suma de triangular de la matriz: 17
PS D:\tec\trabajosProg>
```

Darío Cuauhtémoc Peña Mariano
Tarea 6 programa 3

Funcionamiento:

Este programa usa dos funciones, la primera “contadorPorpalabras” que lo que hace es recibir un string como argumento, esta la descompone en palabras y hace un conteo de cuantas veces aparecieron en el string mostrando los resultados por terminal, la segunda función hace lo mismo que la primera, sin embargo, varía en que la segunda función hace el conteo por caracter. El programa pide una oración al usuario y ejecuta nuestras funciones con la oración del usuario como argumento de ambas.

Pseudocódigo:

```
Inicio
funcion contadorPorpalabras(oracion):

  mostrar("oracion:", oracion)
  oracion<--oracion.split(" ") #[oracion, sin, letras]
  setoracion=set(oracion)

funcion valoresunicosdiccionario (oracionVunicos ):
  key <--0
  diccionarioValores <-- {}
  for i in oracionVunicos:
    diccionarioValores[oracion[key]]<-- 0
    key <--key+1
  regresar diccionarioValores
diccionario <--valoresunicosdiccionario(setoracion)
n<-- 0
for x in oracion:
  if x == oracion[n]:
    diccionario[oracion[n]] <-- 1+diccionario[oracion[n]]
    n<-- n+1
for i in setoracion:
   "oracion"
   mostrar("llave:", i, "----valor:",diccionario[i])
mostrar("---------------")
regresar diccionario
```

```
funcion contadorPorLetras(oracionletras):
   oracion <-- oracionletras
   lista_oracion <-- []
   for letra in oracion:
      if letra not in lista_oracion:
        lista_oracion.agregar(letra)
    oracion_sin_duplicados <-- "".join(lista_oracion)
     mostrar("oracion:", oracionletras)

  setoracion<--oracion_sin_duplicados
listaletrasUnicas<--list(setoracion)
funcion valoresunicosletras(oracionVunicos ):
  key <--0
  diccionarioValores<-- {}
  for i in oracionVunicos:
    diccionarioValores[oracionVunicos[key]]<-- 0
    key <--key+1
  regresar diccionarioValores
diccionario <--valoresunicosletras(listaletrasUnicas)

n<--0
for x in oracionletras:
  if x == oracionletras[n]:
    diccionario[oracionletras[n]] <--1 + diccionario[oracionletras[n]]
    n<--n+1
  for i in oracion_sin_duplicados:
     mostrar("llave:", i, "----valor:",diccionario[i])
  mostrar("----------------")
  regresar diccionario

oracionUsuario<--input("ingrese una oracíón")
contadorPorLetras(oracionUsuario)
contadorPorpalabras(oracionUsuario)
regresar 0
```
Fin

Funcionamiento del programa:

python > tareas > tarea6 > programa3.py > ...

```
1
2
3
4  > def contadorPorpalabras(oracion): ···
38
39
40
41 > def contadorPorLetras(oracionletras): ···
82
83  oracionUsuario= input("ingrese una oracíón")
84  contadorPorLetras(oracionUsuario)
85  contadorPorpalabras(oracionUsuario)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE COMMEN

```
PS D:\tec\trabajosProg> & C:/Users/dari-/AppData/Local/Microsoft/WindowsA
josProg/python/tareas/tarea6/programa3.py
ingrese una oracíónmucho gusto gusto
oracion: mucho gusto gusto
llave: m ----valor: 1
llave: u ----valor: 3
llave: c ----valor: 1
llave: h ----valor: 1
llave: o ----valor: 3
llave:   ----valor: 2
llave: g ----valor: 2
llave: s ----valor: 2
llave: t ----valor: 2
---------------
oracion: mucho gusto gusto
llave: gusto ----valor: 2
llave: mucho ----valor: 1
---------------
PS D:\tec\trabajosProg>
```

Darío Cuauhtémoc Peña Mariano

Tarea 6programa 4

Funcionamiento:

Este programa usa una función, “ordenarMin”, la cual recibe un argumento el cual es una lista, la funcion elige el elemento menor y lo almacena en una nueva lista, si en el argumento hay un elemento o menos regresa la lista creada, en caso contrario se vuelve a llamar a la función pasando como argumento la lista de numeros minimos creada más el elemento minimo del argumento pasado. El programa pide un numero al usuario el cual será el numero de elementos de una lista que se creara, cuyo cada elemento será un número aleatorio del 1 al 100, después esta lista aleatoria se pasará como argumento a nuestra función “ordenarMin”, para finalizar el programa imprimiremos la lista original y la lista ordenada de menor a mayor gracias a nuestra funcion.

Pseudocódigo:

Paquetes: Random

Inicio

```
funcion orenarMin(listaRandom):

  if len(listaRandom) <= 1:
    regresar listaRandom
  elementominimo <-- min(listaRandom)
  nueva_lista <--[x for x in listaRandom if x != elementominimo]
  regresa [elementominimo] + orenarMin(nueva_lista)

n <-- int(input("ingresa numero de elementos de la lista"))
crearListaAleatoria <-- [random.randint(1, 100) for _ in range(n)]
listaOrdenada <-- orenarMin(crearListaAleatoria)

mostrar("Lista original:", crearListaAleatoria)
mostrar("Lista ordenada:", listaOrdenada)

regresar 0
```

Fin

Funcionamiento del programa:

reto1_prog2.py U, M | reto1_prog3.py U, M | programa3.py U, M | *programa4.py U, M*

python > tareas > tarea6 > programa4.py > ...

```python
import random


def orenarMin(listaRandom):

    if len(listaRandom) <= 1:
        return listaRandom
    elementominimo = min(listaRandom)
    nueva_lista = [x for x in listaRandom if x != elementominimo]
    return [elementominimo] + orenarMin(nueva_lista)

n = int(input("ingresa numero de elementos de la lista"))
crearListaAleatoria = [random.randint(1, 100) for _ in range(n)]
listaOrdenada = orenarMin(crearListaAleatoria)

print("Lista original:", crearListaAleatoria)
print("Lista ordenada:", listaOrdenada)
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SQL CONSOLE | COMMENTS | Python

```
PS D:\tec\trabajosProg> & C:/Users/dari-/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:
josProg/python/tareas/tarea6/programa4.py
ingresa numero de elementos de la lista20
Lista original: [72, 94, 21, 72, 38, 14, 61, 65, 69, 3, 5, 54, 56, 81, 6, 28, 64, 58, 1, 43]
Lista ordenada: [1, 3, 5, 6, 14, 21, 28, 38, 43, 54, 56, 58, 61, 64, 65, 69, 72, 81, 94]
PS D:\tec\trabajosProg>
```