

```
1_suma_de_elementos.py.py - C:\Users\User\Desktop\tarea_funcion.py\1_suma_de_elementos.py.py (3.12.4)
File Edit Format Run Options Window Help
lista=[10,7,3,7,2]
suma=0
x=0
while x<len(lista):
    suma=suma+lista[x]
    x=x+1

print(" los numero de la lista son: ")
print(lista)
print(" la suma de los numero es: ")
print(suma)

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\User\Desktop\tarea_funcion.py\1_suma_de_elementos.py.py ==
los numero de la lista son:
[10, 7, 3, 7, 2]
la suma de los numero es:
29
>>>
```

```
2_notas.py.py - C:\Users\User\Desktop\tarea_funcion.py\2_notas.py.py (3.12.4)
File Edit Format Run Options Window Help
#crear una lista con un nombre y dos valores e
lista=["Ana", 13,10]
print("nombre del alumno es:")
print(lista[0])
#dividimos los valores antes de sumarlos para q
promedio= (lista[1])/2 +(lista[2])/2
print("el promedio de sus notas es: ")
print(promedio)

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\User\Desktop\tarea_funcion.py\2_notas.py.py ==
nombre del alumno es:
Ana
el promedio de sus notas es:
11.5
>>>
```

```
3_ingresar_valores.py.py - C:\Users\User\Desktop\tarea_funcion.py\3_ingresar_valores.py.py (3.12.4)
File Edit Format Run Options Window Help
#definimos una lista vacia
lista= []
#disponemos un ciclo de 5 vueltas

for i in range (5):
    valor = int(input("ingrese un numero entero: "))
    lista.append(valor)

print (lista)
```

```
IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: C:\Users\User\Desktop\tarea_funcion.py\3_ingresar_valores.py.py
ingrese un numero entero: 4
ingrese un numero entero: 6
ingrese un numero entero: 9
ingrese un numero entero: 2
ingrese un numero entero: 7
[4, 6, 9, 2, 7]

>>>
```

```
4_tamano_indefinido.py.py - C:\Users\User\Desktop\tarea_funcion.py\4_tamano_indefinido.py.py (3.12.4)
File Edit Format Run Options Window Help
#crear una lista y pedir los valores por teclado
lista=[]
valor = int(input(" ingrese un numero entero o 0 para salir "))
while valor != 0:
    lista.append(valor)
    valor = int(input(" ingrese un numero entero o 0 para salir "))

print ("el tamaño de la lista es: ")
print (len(lista))
```

```
IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: C:\Users\User\Desktop\tarea_funcion.py\4_tamano_indefinido.py.py
ingrese un numero entero o 0 para salir 36
ingrese un numero entero o 0 para salir 4
ingrese un numero entero o 0 para salir 3
ingrese un numero entero o 0 para salir 7
ingrese un numero entero o 0 para salir 8
ingrese un numero entero o 0 para salir 21
ingrese un numero entero o 0 para salir 0
el tamaño de la lista es:
6

>>>
```

```
5_Menor_de_lista.py.py - C:\Users\User\Desktop\tarea_funcion.py\5_Menor_de_lista.py.py (3.12.4)
File Edit Format Run Options Window Help
#crear una lista de 5 pocisiones pidiendo los valores
lista = []
for x in range (5):
    valor=int(input("ingrese un numero entero "))
    lista.append(valor)

menor = lista[0]
posicion = 0

for x in range (1,5):
    if lista[x]<menor:
        menor= lista[x]
        posicion = x

print (" lista completa ")
print (lista)
print (" el numero menor es: ")
print (menor)
print (" se encuentra en la posicion ")
print (posicion)

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MS
C v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more inf
ormation.
>>>
= RESTART: C:\Users\User\Desktop\tarea_funcion.py\5_Menor_de_li
sta.py.py
ingrese un numero entero 4
ingrese un numero entero 6
ingrese un numero entero 2
ingrese un numero entero 8
ingrese un numero entero 9
lista completa
[4, 6, 2, 8, 9]
el numero menor es:
2
se encuentra en la posicion
2
>>>
```

```
6_listas_paralelas.py.py - C:\Users\User\Desktop\tarea_funcion.py\6_listas_paralelas.py.py (3.12.4)
File Edit Format Run Options Window Help
#programa permite cargar 5 nombres de personas con sus edades
nombres =[]
edades = []

for x in range (5):
    nombre = input ("ingrese el nombre ")
    nombres.append(nombre)

    edad= int(input(" ingrese la edad de dicha persona "))
    edades.append(edad)

print (" nombres de las personas mayores de edad ")
for x in range (5):
    if edades [x] >= 18:
        print (nombres [x])

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MS
C v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\User\Desktop\tarea_funcion.py\6_listas_paralelas.py.py
ingrese el nombre Jorge
ingrese la edad de dicha persona 13
ingrese el nombre Manuel
ingrese la edad de dicha persona 23
ingrese el nombre Romeo
ingrese la edad de dicha persona 17
ingrese el nombre Fernando
ingrese la edad de dicha persona 34
ingrese el nombre Luis
ingrese la edad de dicha persona 21
nombres de las personas mayores de edad
Manuel
Fernando
Luis
>>>
```

```
7_tuplas.py - C:\Users\User\Desktop\tarea_funcion.py\7_tuplas.py (3.12.4)
File Edit Format Run Options Window Help
#definir varias tuplas e imprimir sus elementos
tupla = (1,2,3,)
fecha = (25, "diciembre", 2016)
punto = (10,2)
personas = ("pedro", "pablo", 43)
print (tupla)
print (fecha)
print (punto)
print (personas)

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
= RESTART: C:\Users\User\Desktop\tarea_funcion.py\7_tuplas.p
Y.PY
(1, 2, 3)
(25, 'diciembre', 2016)
(10, 2)
('pedro', 'pablo', 43)
>>>
```

```
8_lista_y_tuplas.py - C:\Users\User\Desktop\tarea_funcion.py\8_lista_y_tuplas.py (3.12.4)
File Edit Format Run Options Window Help
#definir una tupla con tres valores enteros conver
#y convertirla en tupla
fechatupla1=(25, 12, 2016)
print ("imprimimos la primera tupla ")
print (fechatupla1)

#convertimos la tupla en lista
fechalista1=list(fechatupla1)
print (" imprimimos la lista copiada de la tupla ")
print (fechalista1)
#modificamos la lista
fechalista1[0]=31
print (" imprimimos la lista modificada ")
print (fechalista1)
#convertimos la lista en tupla
fechatupla2=tuple(fechalista1)
print (" imprimimos la nueva tupla ")
print (fechatupla2)

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more info
rmation.
>>>
= RESTART: C:\Users\User\Desktop\tarea_funcion.py\8_lista_y_tupl
as.py.py
imprimimos la primera tupla
(25, 12, 2016)
imprimimos la lista copiada de la tupla
[25, 12, 2016]
imprimimos la lista modificada
[31, 12, 2016]
imprimimos la nueva tupla
(31, 12, 2016)
>>>
```

```
9_tuplas_y_funciones.py.py - C:\Users\User\Desktop\tarea_funcion.py\9_tuplas_y_funciones.py.py (3.12.4)
File Edit Format Run Options Window Help

# Lista para almacenar los empleados y sus sueldos
empleados = []

# Función 1: Carga de empleados
def cargar_empleados():
    for i in range(5):
        nombre = input(f"Ingrese el nombre del empleado {i+1}: ")
        sueldo = float(input(f"Ingrese el sueldo de {nombre}: "))
        empleados.append((nombre, sueldo))

# Función 2: Imprimir empleados y sus sueldos
def imprimir_empleados():
    print("\nLista de empleados y sus sueldos:")
    for nombre, sueldo in empleados:
        print(f"Empleado: {nombre}, Sueldo: {sueldo}")

# Función 3: Nombre del empleado con sueldo mayor
def empleado_sueldo_mayor():
    empleado_max = max(empleados, key=lambda x: x[1]) # Usamos la
    print(f"El empleado con el sueldo mayor es {empleado_max[0]}")

# Función 4: Cantidad de empleados con sueldo menor a 1000
def empleados_sueldo_menor_1000():
    cantidad = sum(1 for _, sueldo in empleados if sueldo < 1000)
    print(f"\nLa cantidad de empleados con sueldo menor a 1000 es: ")

# Programa principal
cargar_empleados()
imprimir_empleados()
empleado_sueldo_mayor()
empleados_sueldo_menor_1000()

IDLE Shell 3.12.4
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: C:\Users\User\Desktop\tarea_funcion.py\9_tuplas_y_funciones.py.py
Ingrese el nombre del empleado 1: Jorge
Ingrese el sueldo de Jorge: 800
Ingrese el nombre del empleado 2: Byron
Ingrese el sueldo de Byron: 1230
Ingrese el nombre del empleado 3: Santos
Ingrese el sueldo de Santos: 525
Ingrese el nombre del empleado 4: sonia
Ingrese el sueldo de sonia: 3000
Ingrese el nombre del empleado 5: Fernando
Ingrese el sueldo de Fernando: 2315

Lista de empleados y sus sueldos:
Empleado: Jorge, Sueldo: 800.0
Empleado: Byron, Sueldo: 1230.0
Empleado: Santos, Sueldo: 525.0
Empleado: sonia, Sueldo: 3000.0
Empleado: Fernando, Sueldo: 2315.0

El empleado con el sueldo mayor es sonia con un sueldo de 3000.0

La cantidad de empleados con sueldo menor a 1000 es: 2

>>>
```

```
almacenar_articulos.py.py - C:\Users\User\Desktop\tarea_funcion.py\almacenar_articulos.py.py (3.12.4)
File Edit Format Run Options Window Help

#crear un diccionario ue permita almacenar articulos
#utilizar como clave el nombre del producto y como valor el
#desarrollar las funciones de
#imprimir en forma completa el diccionario
#imprimir solo los articulos con precios superior a 100

def cargar():
    productos={}
    for x in range (5):
        nombre=input(" ingrese el nombre del producto ")
        precio=int(input(" ingrese el precio del mismo "))
        productos[nombre]=precio
    return productos

def imprimir(productos):
    print("listado de todos los articulos ")
    for nombre in productos:
        print (nombre, productos[nombre])

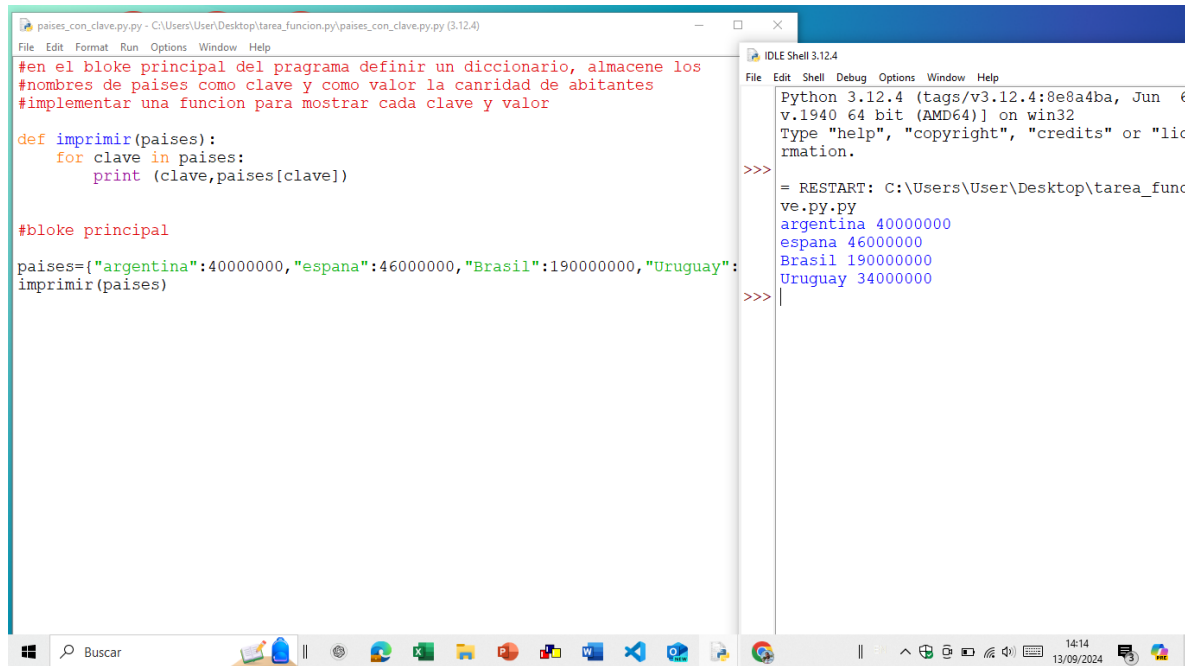
def imprimir_mayor100(productos):
    print("listado de productos mayores a 100 ")
    for nombre in productos:
        if productos[nombre]>100:
            print(nombre)

#bloke principal
productos=cargar()
imprimir(productos)
imprimir_mayor100(productos)

IDLE Shell 3.12.4
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
= RESTART: C:\Users\User\Desktop\tarea_funcion.py\almacenar_articulos.py.py
ingrese el nombre del producto azucar
ingrese el precio del mismo 5
listado de todos los articulos
azucar 5
listado de productos mayores a 100

>>>
```



The image shows a screenshot of a Python IDE with two windows. The left window, titled 'países\_con\_clave.py', contains a Python script. The script has three comments in Spanish: '#en el bloque principal del programa definir un diccionario, almacene los nombres de países como clave y como valor la cantidad de habitantes' and '#implementar una función para mostrar cada clave y valor'. It defines a function 'imprimir(paises):' that iterates over the dictionary and prints each key-value pair. In the main block, a dictionary 'paises' is created with four entries: Argentina (40,000,000), Spain (46,000,000), Brazil (190,000,000), and Uruguay (34,000,000). The 'imprimir' function is then called. The right window, titled 'IDLE Shell 3.12.4', shows the output of the script: the path to the script file, followed by the four country names and their corresponding population values, each on a new line. The Windows taskbar at the bottom shows the date as 13/09/2024 and the time as 14:14.

```
países_con_clave.py - C:\Users\User\Desktop\tarea_funcion.py\países_con_clave.py (3.12.4)
File Edit Format Run Options Window Help

#en el bloque principal del programa definir un diccionario, almacene los
#nombres de países como clave y como valor la cantidad de habitantes
#implementar una función para mostrar cada clave y valor

def imprimir(paises):
    for clave in paises:
        print (clave,paises[clave])

#bloque principal

paises={"argentina":40000000,"espana":46000000,"Brasil":190000000,"Uruguay":
imprimir(paises)

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help

Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6
v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "lic
rmation.

>>>
= RESTART: C:\Users\User\Desktop\tarea_func
ve.py.py
argentina 40000000
espana 46000000
Brasil 190000000
Uruguay 34000000
>>> |
```