

```
Python program to check if year is a leap year or not

year = 2000

# To get year (integer input) from the users
year = int(input("Enter a year: "))

# divided by 100 means century year (ending with 00)
# century year divided by 400 is leap year
if (year % 400 == 0) and (year % 100 == 0):
    print("{} is a leap year".format(year))

# not divided by 100 means not a century year
# year divided by 4 is a leap year
elif (year % 4 == 0) and (year % 100 != 0):
    print("{} is a leap year".format(year))

# if not divided by both 400 (century year) and 4 (not century year)
# year is not leap year
else:
    print("{} is not a leap year".format(year))
```

```
Python 3.10.1 (tags/v3.10.1.32ed64e, Dec 6 2021, 18:10:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/usuario/Desktop/Tipos Compuestos/ASO EJERCIS70.py =====
2000 is a leap year
>>>
```

```
Python program to find the largest number among the three input numbers

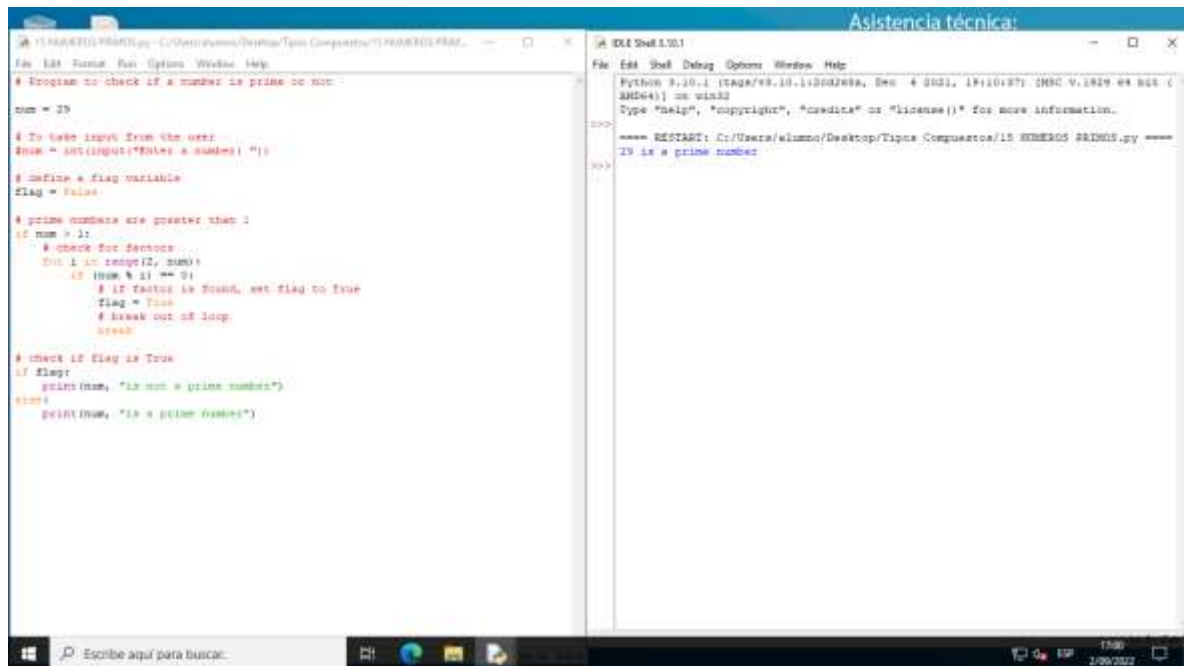
# change the values of num1, num2 and num3
# for a different result
num1 = 10
num2 = 14
num3 = 12

# uncomment following lines to take three numbers from users
#num1 = float(input("Enter first number: "))
#num2 = float(input("Enter second number: "))
#num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
    largest = num1
elif (num2 >= num1) and (num2 >= num3):
    largest = num2
else:
    largest = num3

print("The largest number is", largest)
```

```
Python 3.10.1 (tags/v3.10.1.110b686a, Dec 6 2021, 18:10:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/usuario/Desktop/Tipos Compuestos/OPERACIONES LOGICAS.py =====
The largest number is 14
>>>
```



Asistencia técnica:

Python 3.10.1 (tags/v3.10.11.02600, Dec 8 2021, 19:10:17) [AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
==== RESTART: C:/Users/usuario/Desktop/Tips Computacion/PRIMOS EX US SARGO.py ====
Prime numbers between 900 and 1000 are:
907
911
919
929
937
941
947
953
967
971
977
983
991
997
1000

Python program to display all the prime numbers within an interval

lower = 900
upper = 1000

def main("Prime numbers between", lower, "and", upper, "are"):

 for num in range(lower, upper + 1):
 # all prime numbers are greater than 1
 if num > 1:
 for i in range(2, num):
 if (num % i) == 0:
 break
 else:
 print(num)

Asistencia técnica:

Python 3.10.1 (tags/v3.10.11.02600, Dec 8 2021, 19:10:17) [AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
==== RESTART: C:/Users/usuario/Desktop/Tips Computacion/OPERACIONES STRING.py ====
Hola Mundo
Hola Hola Mundo
Hola Mundo
10
5
-1
Hola mundo
H0p1saa MUND0

OPERACIONES STRING.py - C:/Users/usuario/Desktop/Tips Computacion/OPERACIONES ST...
File Edit Format Run Options Window Help

#Concatenar Cadena
mensaje1 = "Hola" + " " + "Mundo"
print(mensaje1)

#Multiplicar Cadena
mensaje2a = "Hola" * 3
mensaje2b = "Mundo"
print(mensaje2a + mensaje2b)

#Reemplazar Cadena
mensaje3 = "Hola"
mensaje3 += " " + "Mundo"
print(mensaje3)

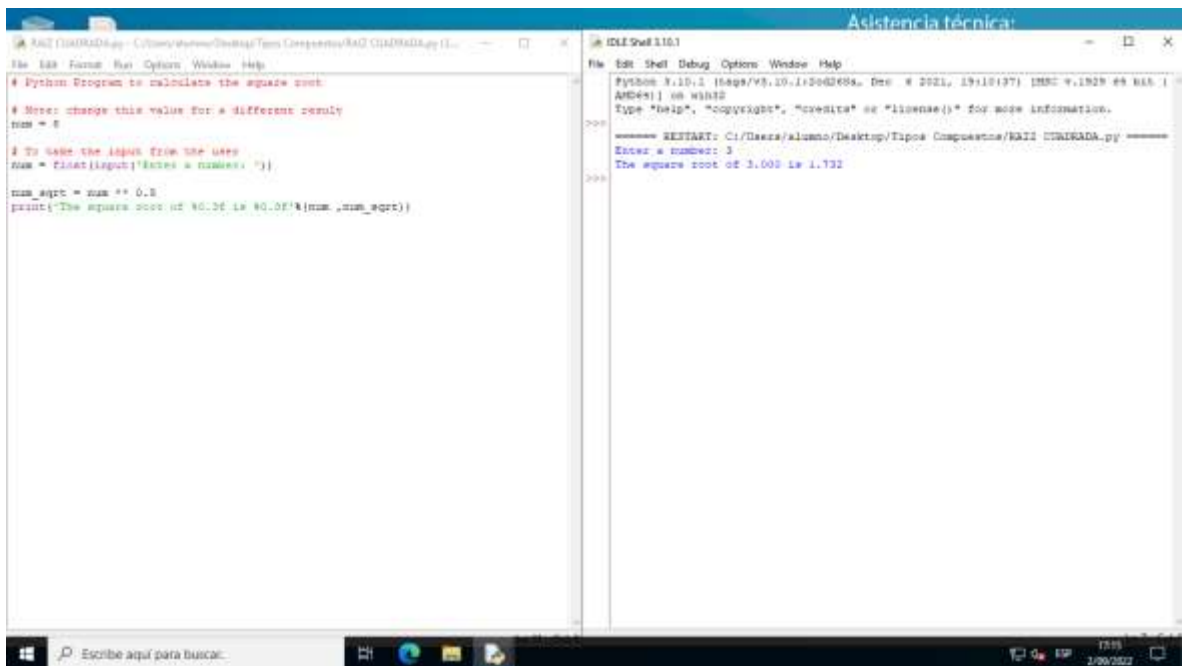
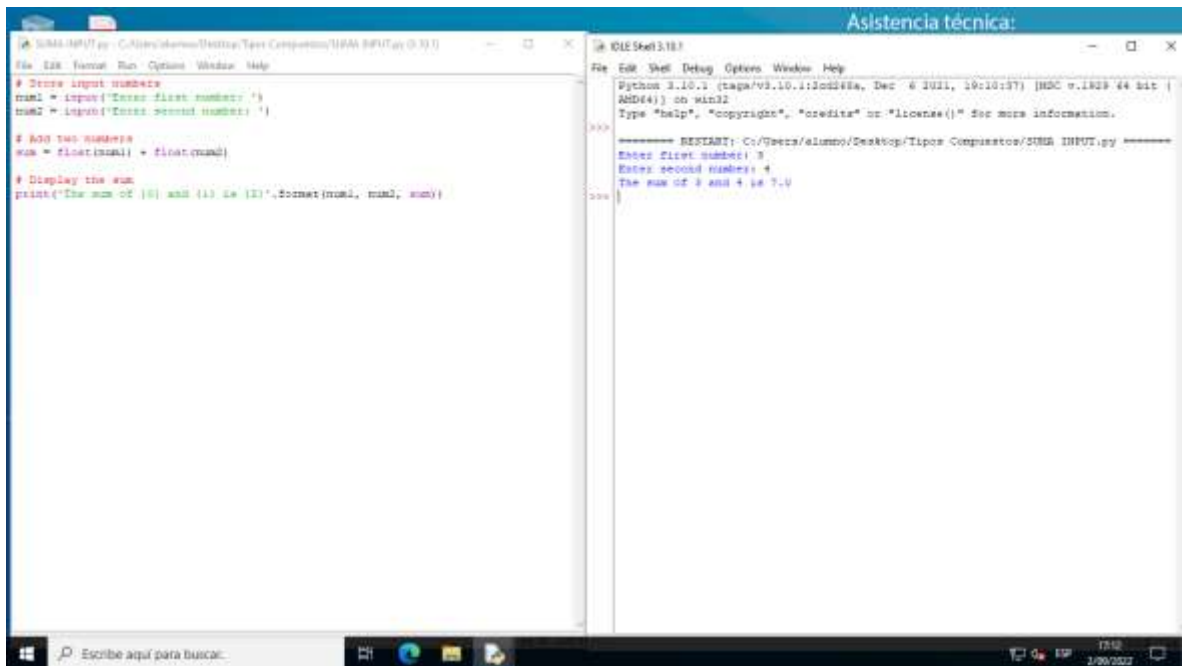
#Iterar String
mensaje4 = "Hola" + " " + "Mundo"
print(len(mensaje4))

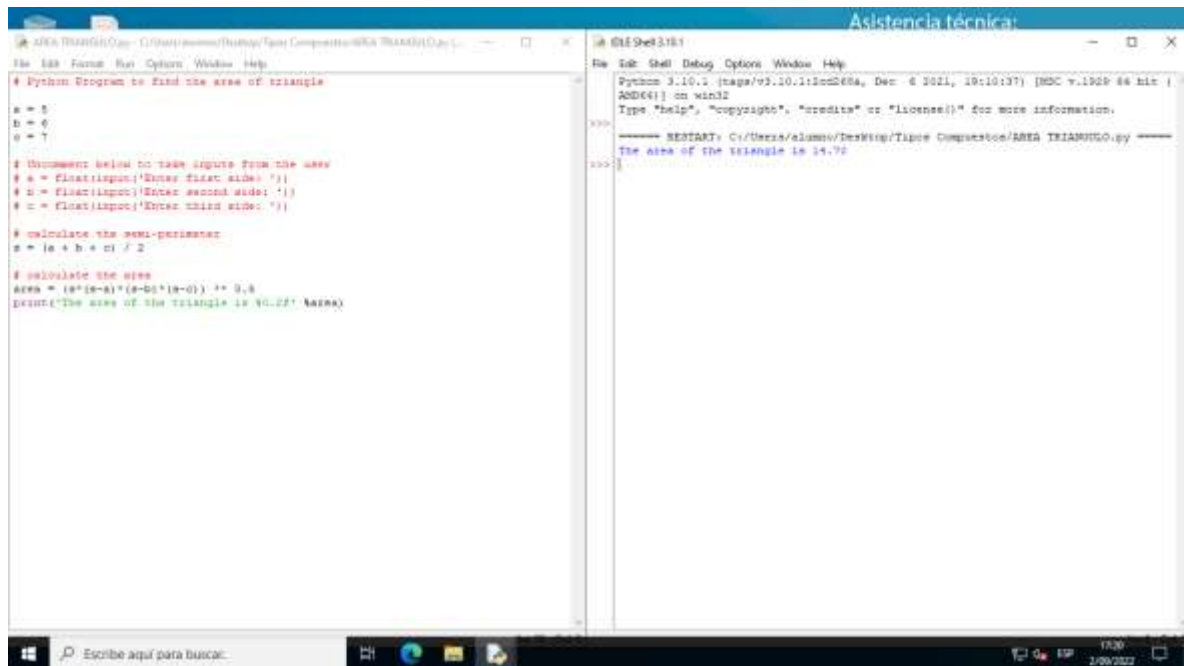
#Buscar
mensaje5 = "Hola Mundo"
mensaje5a = mensaje5.find("Mundo")
print(mensaje5a)

#Reemplazar
mensaje6 = "Hola Mundo"
mensaje6a = mensaje6.find("Mundo")
print(mensaje6a)

#Reemplazar
mensaje7 = "HOLA MUNDO"
mensaje7a = mensaje7.lower()
print(mensaje7a)

#Reemplazar
mensaje8 = "HOLA MUNDO"
mensaje8a = mensaje8.replace("L", "piza")
print(mensaje8a)





```

C:\Users\alumno\Desktop\Tipos Compuestos\EJERCICIO 10...
File Edit Format Run Options Window Help
# Solve the quadratic equation ax**2 + bx + c = 0
# Import complex math module
import math

a = 1
b = 5
c = 6

# Calculate the discriminant
d = (b**2) - (4*a*c)

# Find the solutions
sol1 = (-b-math.sqrt(d))/(2*a)
sol2 = (-b+math.sqrt(d))/(2*a)

print("The solution are {} and {}".format(sol1,sol2))

```

Asistencia técnica:

```

Python 3.10.1 (tags/v3.10.1.12ed208a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/alumno/Desktop/Tipos Compuestos/EJERCICIO 10... ===
The solution are (-2+0j) and (-3+0j)
>>>

```

```

C:\Users\alumno\Desktop\Tipos Compuestos\INTERCAMBIO DE...
File Edit Format Run Options Window Help
# Python program to swap two variables

x = 5
y = 10

print("x = ", x)
print("y = ", y)

# To take inputs from the user
#x = input("Enter value of x: ")
#y = input("Enter value of y: ")

# create a temporary variable and swap the values
temp = x
x = y
y = temp

print("The value of x after swapping: {}".format(x))
print("The value of y after swapping: {}".format(y))

#without using temporary variable

x, y = y, x
print("The value of x after swapping without temporary: {}".format(x))
print("The value of y after swapping without temporary: {}".format(y))

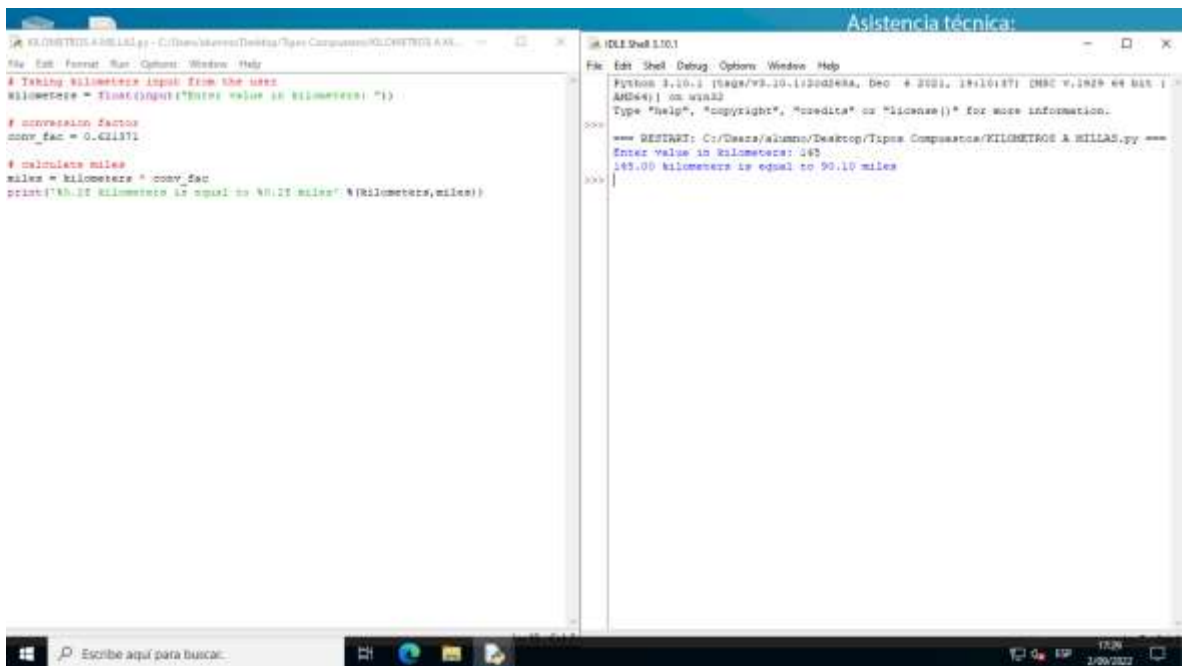
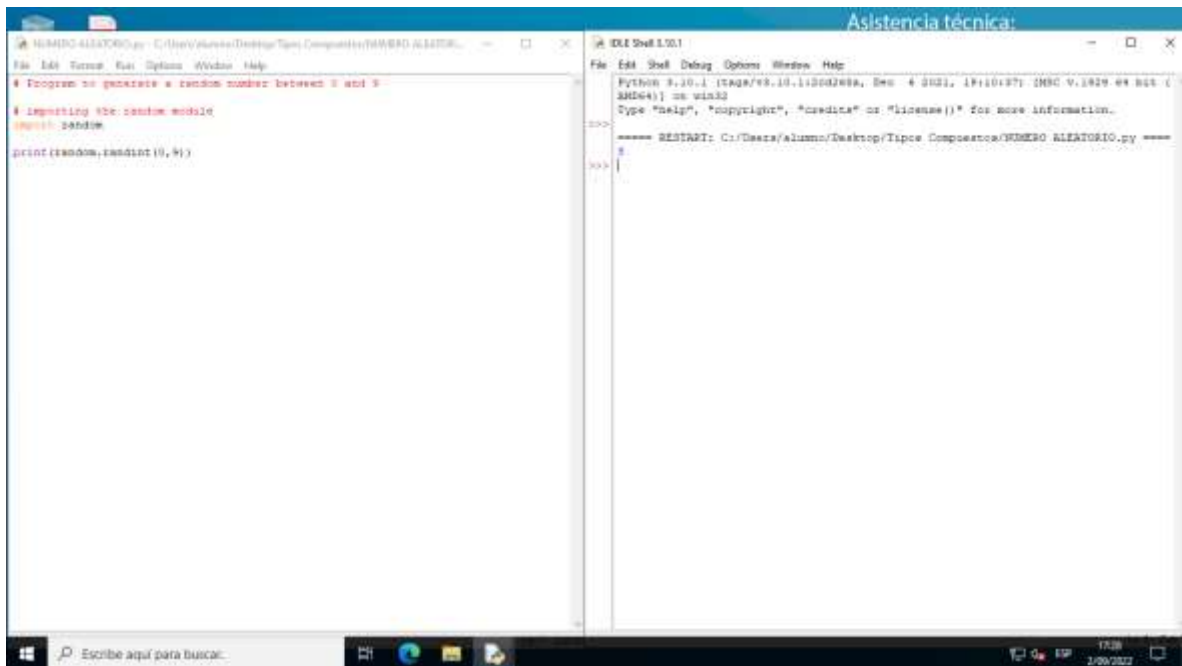
```

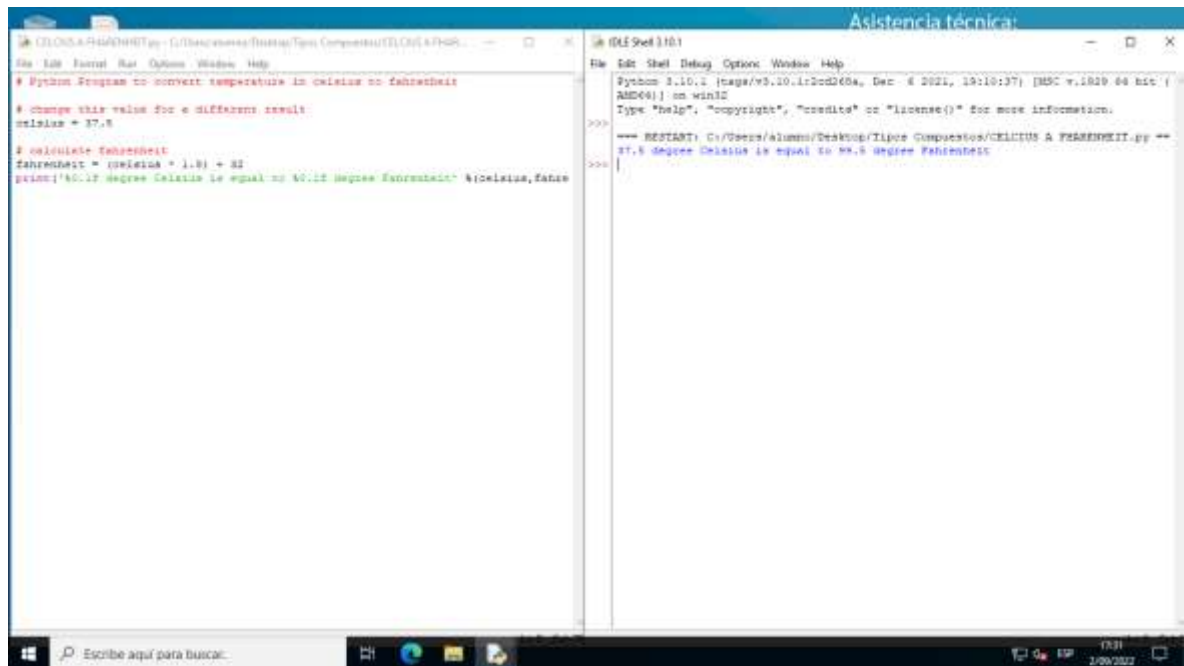
Asistencia técnica:

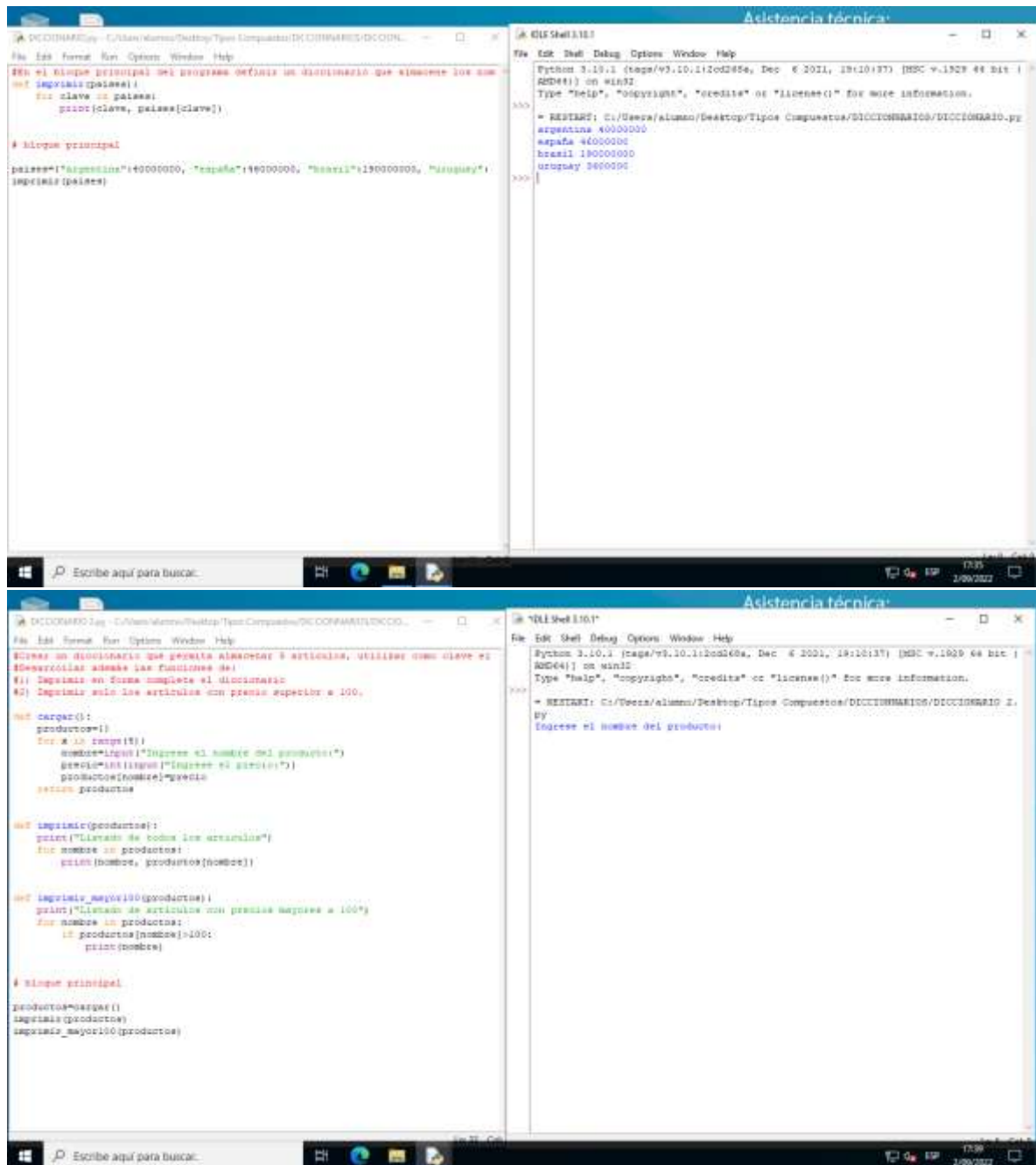
```

Python 3.10.1 (tags/v3.10.1.12ed208a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/alumno/Desktop/Tipos Compuestos/INTERCAMBIO DE VARIABLES.py ===
x = 5
y = 10
The value of x after swapping: 10
The value of y after swapping: 5
The value of x after swapping without temporary: 5
The value of y after swapping without temporary: 10
>>>

```





```
DICCIONARIO 3.py - C:\Users\alumno\Desktop\Tipos Computacionales\PROYECTOS\PROYECTOS...
File Edit Format Run Options Window Help
#Desarrollar una aplicación que nos permita crear un diccionario inglés/castellano
#Crear las siguientes funciones:
#1) Cargar el diccionario.
#2) Listado completo del diccionario.
#3) Imprimir por teclado una palabra en inglés y si existe en el diccionario mostrarla.

def cargar():
    diccionario={}
    continua="s"
    while continua=="s":
        ing=input("Ingrese palabra en castellano:")
        esp=input("Ingrese palabra en inglés:")
        diccionario[ing]=esp
        continua=input("Quiere cargar otra palabra:[s/n]")
    return diccionario

def imprimir(diccionario):
    print("Listado completo del diccionario:")
    for ingles in diccionario:
        print(ingles,diccionario[ingles])

def consulta_palabra(diccionario):
    pal=input("Ingrese la palabra en inglés a consultar:")
    if pal in diccionario:
        print("En castellano significa:",diccionario[pal])

# Bloque principal
diccionario=cargar()
imprimir(diccionario)
consulta_palabra(diccionario)
```

```
Python 3.10.1 (tags/v3.10.1:20d2466, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/alumno/Desktop/Tipos Computacionales/DICCIONARIOS/DICCIONARIO 3.py
Ingrese palabra en castellano:COMIDA
Ingrese palabra en inglés:FOOD
Quiere cargar otra palabra:[s/n]n
Listado completo del diccionario
FOOD COMIDA
Ingrese la palabra en inglés a consultar:FOOD
En castellano significa: COMIDA
>>>
```

```
DICCIONARIO DE TIPLAS Y LISTAS.py - C:\Users\alumno\Desktop\Tipos Computacionales\PROYECTOS\PROYECTOS...
File Edit Format Run Options Window Help
#Desarrollar una aplicación que nos permita crear un diccionario inglés/castellano
#Crear las siguientes funciones:
#1) Cargar el diccionario.
#2) Listado completo del diccionario.
#3) Imprimir por teclado una palabra en inglés y si existe en el diccionario mostrarla.

def cargar():
    productos=[]
    continua="s"
    while continua=="s":
        codigo=int(input("Ingrese el código del producto:"))
        descripcion=input("Ingrese la descripción:")
        precio=float(input("Ingrese el precio:"))
        stock=int(input("Ingrese el stock actual:"))
        productos.append([descripcion,precio,stock])
        continua=input("Quiere cargar otro producto:[s/n]")
    return productos

def imprimir(productos):
    print("Listado completo de productos:")
    for codigo in productos:
        print(codigo,productos[codigo][0],productos[codigo][1],productos[codigo][2])

def consulta(productos):
    codigo=int(input("Ingrese el código de artículo a consultar:"))
    if codigo in productos:
        print(productos[codigo][0],productos[codigo][1],productos[codigo][2])

def listado_stock_cero(productos):
    print("Listado de artículos con stock en cero:")
    for codigo in productos:
        if productos[codigo][2]==0:
            print(codigo,productos[codigo][0],productos[codigo][1],productos[codigo][2])

# Bloque principal
productos=cargar()
imprimir(productos)
consulta(productos)
listado_stock_cero(productos)
```

```
Python 3.10.1 (tags/v3.10.1:20d2466, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/alumno/Desktop/Tipos Computacionales/DICCIONARIOS/DICCIONARIO DE TIPLAS Y LISTAS.py
Ingrese el código del producto:123
Ingrese la descripción:COMIDA PARA GATO
Ingrese el precio:7.50
Ingrese el stock actual:300
Quiere cargar otro producto:[s/n]n
Ingrese el código del producto:123
Ingrese la descripción:COMIDA PARA PERRO
Ingrese el precio:8.00
Ingrese el stock actual:120
Quiere cargar otro producto:[s/n]n
Listado completo de productos:
123 COMIDA PARA GATO 7.5 300
123 COMIDA PARA PERRO 8.0 120
Ingrese el código de artículo a consultar:123
COMIDA PARA GATO 7.5 300
Listado de artículos con stock en cero:
>>>
```

```
DOCCIONARIO DE TUPLAS Y LISTAS.py: C:\Users\alumno\Desktop\Tipos Compuestos\DOCCIONARIO DE TUPLAS Y LISTAS 2.py
File Edit Format Run Options Window Help

#Comprobamos una agenda. Hallar un diccionario cuyo clave sea la fecha. Permitir
#Implementar las siguientes funciones:
#1) Carga de datos en la agenda.
#2) Listado completo de la agenda.
#3) Consulta de una fecha.

def cargar():
    agenda={}
    continua="s"
    while continua=="s":
        fecha=input("Ingrese la fecha con formato dd/mm/aa:")
        continua="n"
        lista=[]
        while continua=="s":
            hora=input("Ingrese la hora de la actividad con formato hh:mm ")
            actividad=input("Ingrese la descripción de la actividad:")
            lista.append((hora,actividad))
            actividad=input("Ingrese otra actividad para la misma fecha[s/n]:")
        agenda[fecha]=lista
        continua=input("Ingrese otra fecha[s/n]:")
    return agenda

def imprimir(agenda):
    print("\nListado completo de la agenda:")
    for fecha in agenda:
        print("Para la fecha:",fecha)
        for hora,actividad in agenda[fecha]:
            print(hora,actividad)

def consulta_fecha(agenda):
    fecha=input("Ingrese la fecha que desea consultar:")
    if fecha in agenda:
        for hora,actividad in agenda[fecha]:
            print(hora,actividad)
    else:
        print("No hay actividades agendadas para dicha fecha.")
```

[illegible]

Asistencia técnica:

LISTA ELIMINAR.py - C:\Users\alumno\Desktop\Tipos Computacionales\PROGRAMAS\LISTA ELIMINAR.py

File Edit Format Run Options Window Help

#Crea una lista por asignación con 3 valores. Elimina el primero, el tercero y la lista=[10, 20, 30, 40, 50]

print(lista)

lista.pop(0)

lista.pop(1)

lista.pop(2)

print(lista)

Python 3.10.1 (tags/v3.10.1:3cd208a, Dec 6 2021, 19:10:37) [AMD64] on win32

Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:\Users\alumno\Desktop\Tipos Computacionales\PROGRAMAS\LISTA ELIMINAR.py

[10, 20, 30, 40, 50]

[20, 40]

Asistencia técnica:

LISTA ELIMINAR 2.py - C:\Users\alumno\Desktop\Tipos Computacionales\PROGRAMAS\LISTA ELIMINAR 2.py

File Edit Format Run Options Window Help

#Crea una lista y almacena 10 enteros pedidos por teclado. Elimina todos los lista=[]

for x in range(10):

valor=int(input("Ingrese valor:"))

lista.append(valor)

print(lista)

posicion=0

while posicion<len(lista):

if lista[posicion]==6:

lista.pop(posicion)

else:

posicion=posicion+1

print(lista)

Python 3.10.1 (tags/v3.10.1:3cd208a, Dec 6 2021, 19:10:37) [AMD64] on win32

Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:\Users\alumno\Desktop\Tipos Computacionales\LISTAS\LISTAS ELIMINAR 2.py

Ingrese valor:3

Ingrese valor:4

Ingrese valor:5

Ingrese valor:3

Ingrese valor:3

Ingrese valor:4

Ingrese valor:5

Ingrese valor:4

Ingrese valor:6

Ingrese valor:4

[3, 4, 3, 3, 2, 4, 3, 4, 2, 4]

[3, 4, 3, 3, 4, 4, 4]

Asistencia técnica:

Python 3.10.1 (tags/v3.10.1.20d269a, Dec 6 2021, 18:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

== RESTART: C:/Users/usuario/Desktop/Tipos Compuestos/LISTAS/LISTA TAMAÑO INDEFINIDO.py
Ingresar valor (0 para finalizar):3
Ingresar valor (0 para finalizar):4
Ingresar valor (0 para finalizar):5
Ingresar valor (0 para finalizar):3
Ingresar valor (0 para finalizar):0
Tamaño de la lista:
4

LISTA TAMAÑO INDEFINIDO.py - C:/Users/usuario/Desktop/Tipos Compuestos/LISTAS/LISTA TAMAÑO INDEFINIDO.py

File Edit Format View Options Window Help

```
#Realizar la carga de valores enteros por teclado, almacenarlos en una lista. El  
lista=[]  
valor=0  
while valor!=0:  
    valor=int(input("Ingresar valor (0 para finalizar):"))  
    lista.append(valor)  
print("Tamaño de la lista:")  
print(len(lista))
```

Asistencia técnica:

Python 3.10.1 (tags/v3.10.1.20d269a, Dec 6 2021, 18:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

== RESTART: C:/Users/usuario/Desktop/Tipos Compuestos/LISTAS/LISTA TAMAÑO INDEFINIDO.py ==
[[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]

1

1
2
3

1
2
3
4
5
6
7
8
9
10
11
12

13

LISTA DE LISTAS.py - C:/Users/usuario/Desktop/Tipos Compuestos/LISTAS/LISTA DE LISTAS.py

File Edit Format View Options Window Help

```
#Crea una lista por asignación. La lista tiene que tener cuatro elementos. Cada  
#elemento una lista con elementos almacenados de diferentes modos.  
  
lista=[1,2,3], [4,5,6], [7,8,9], [10,11,12]  
  
# imprimimos la lista completa  
print(lista)  
print("-----")  
# imprimimos la primera componente  
print(lista[0])  
print("-----")  
# imprimimos la primera componente de la lista contenida  
# en la primera componente de la lista principal  
print(lista[0][0])  
print("-----")  
# imprimimos con un for la lista contenida en la primera componente  
for x in range(len(lista[0])):  
    print(lista[0][x])  
print("-----")  
# imprimimos cada elemento entero de cada lista contenida en la lista  
for x in range(len(lista)):  
    for y in range(len(lista[x])):  
        print(lista[x][y])
```

LISTAS DE LISTAS 3.py

File Edit Format Run Options Window Help

```
#Crea una lista por asignación. La lista tiene que tener 5 elementos. Cada elemento
#tiene todos los valores de las listas.
lista=[1, 1,1, 1,2,3), 1,2,3,4), 1,2,3,4,5), 1,2,3,4,5,6]]

suma=0
for k in range(len(lista)):
    for x in range(len(lista[k])):
        suma=suma+lista[k][x]
print(suma)
```

IDE Shell 3.10.1

File Edit Shell Debug Options Window Help

Python 3.10.1 (tags/v3.10.1:12a046a, Dec 4 2021, 19:10:37) [AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/alumno/Desktop/Tipos Computacion/LISTAS/LISTA DE LISTAS 3.py
>>>

LISTAS DE LISTAS 3.py

File Edit Format Run Options Window Help

```
#Crea una lista por asignación. La lista tiene que tener 2 elementos. Cada elemento
#tiene la suma de cada lista secundaria de la lista principal.
lista=[1,2,3,4,1,1), 1,2,3,4,2,2)]

suma1=lista[0][0]+lista[0][1]+lista[0][2]+lista[0][3]+lista[0][4]
print(suma1)
suma2=lista[1][0]+lista[1][1]+lista[1][2]+lista[1][3]+lista[1][4]
print(suma2)
print("-----")

suma1=0
for k in range(len(lista[0])):
    suma1=suma1+lista[0][k]
suma2=0
for k in range(len(lista[1])):
    suma2=suma2+lista[1][k]
print(suma1)
print(suma2)
print("-----")

for k in range(len(lista)):
    suma=0
    for x in range(len(lista[k])):
        suma=suma+lista[k][x]
    print(suma)
```

IDE Shell 3.10.1

File Edit Shell Debug Options Window Help

Python 3.10.1 (tags/v3.10.1:12a046a, Dec 4 2021, 19:10:37) [AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/alumno/Desktop/Tipos Computacion/LISTAS/LISTA DE LISTAS 3.py
5
10

5
10

5
10
>>>

Python 3.10.1 (tags/v3.10.1.2042066, Dec 6 2021, 19:10:57) [AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information:

== RESTART: C:/Users/usuario/Desktop/Tipos Computacionales/LISTA/LISTA PARALELA.py ==
Ingresar el nombre de la persona:EVAN
Ingresar la edad de dicha persona:30
Ingresar el nombre de la persona:ALEJANDRO
Ingresar la edad de dicha persona:30
Ingresar el nombre de la persona:JOSE
Ingresar la edad de dicha persona:25
Ingresar el nombre de la persona:CARLOS
Ingresar la edad de dicha persona:21
Ingresar el nombre de la persona:LUIS
Ingresar la edad de dicha persona:18
Nombre de las personas mayores de edad:
EVAN
ALEJANDRO
JOSE
CARLOS
LUIS

#Desarrollar un programa que permita cargar 3 nombres de personas y sus edades en
nombres=[]
edades=[]
for i in range(3):
 nom=input("Ingresar el nombre de la persona:")
 nombres.append(nom)
 ed=int(input("Ingresar la edad de dicha persona:"))
 edades.append(ed)

print("Nombre de las personas mayores de edad:")
for i in range(3):
 if edades[i]>18:
 print(nombres[i])

Python 3.10.1 (tags/v3.10.1.2042066, Dec 6 2021, 19:10:57) [AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information:

== RESTART: C:/Users/usuario/Desktop/Tipos Computacionales/LISTA/LISTA MAYOR O MENOR.py ==
Ingresar valor:3
Ingresar valor:4
Ingresar valor:5
Ingresar valor:6
Ingresar valor:7
Lista completa
[3, 4, 5, 6, 7]
Mayor de la lista
7

#Cargar y cargar una lista con 5 elementos. Implementar un algoritmo que identifique
lista=[]
for i in range(5):
 valor=int(input("Ingresar valor:"))
 lista.append(valor)

mayor=lista[0]
for i in range(1,5):
 if lista[i]>mayor:
 mayor=lista[i]

print("Lista completa")
print(lista)
print("Mayor de la lista")
print(mayor)

```

C:\Users\alumno\Desktop\Tipos Computacionales\LISTAS\MENOR DE LA LISTA.py
File Edit Format Run Options Window Help
#Crea y carga una lista con 5 enteros por teclado. Implementar un algoritmo que
lista=[]
for x in range(5):
    valor=int(input("Ingrese valor:"))
    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("Menor de la lista")
print(menor)
print("Posición del menor en la lista")
print(posicion)

Asistencia técnica:
Python 3.10.1 (tags/v3.10.1:2002000, Dec 6 2021, 19:10:07) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/alumno/Desktop/Tipos Computacionales/LISTAS/MENOR DE LA LISTA.py =====
Ingrese valor:2
Ingrese valor:3
Ingrese valor:4
Ingrese valor:5
Ingrese valor:6
Lista completa
[2, 3, 4, 5, 6]
Menor de la lista
2
Posición del menor en la lista
0
>>>

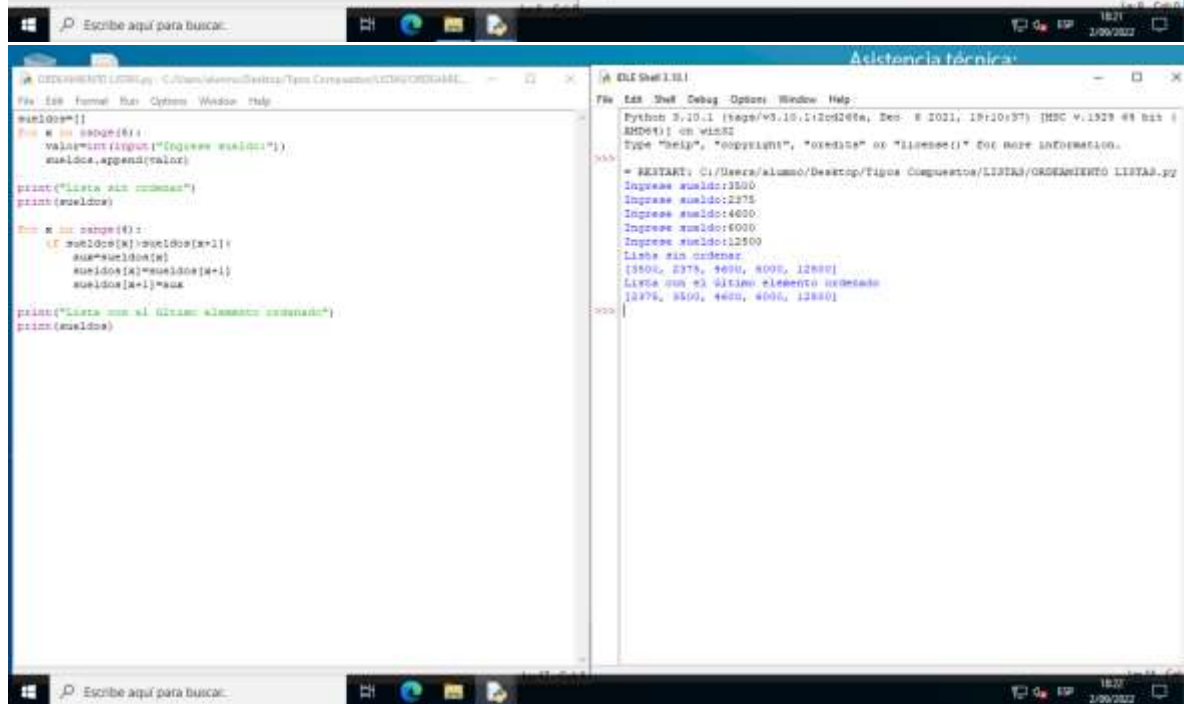
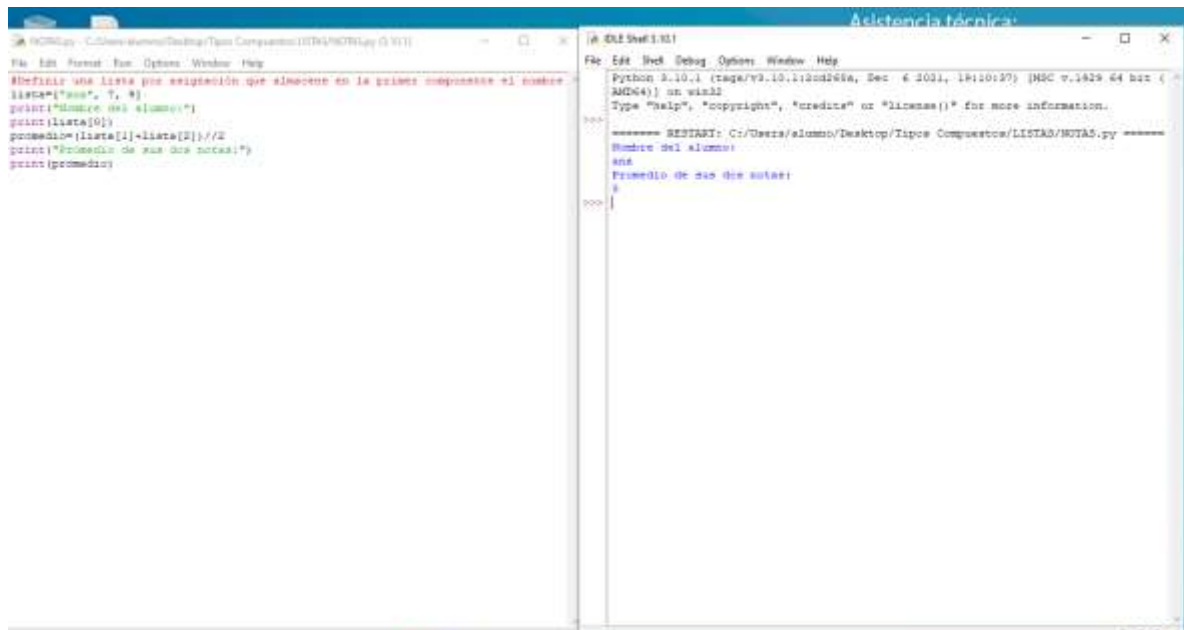
```

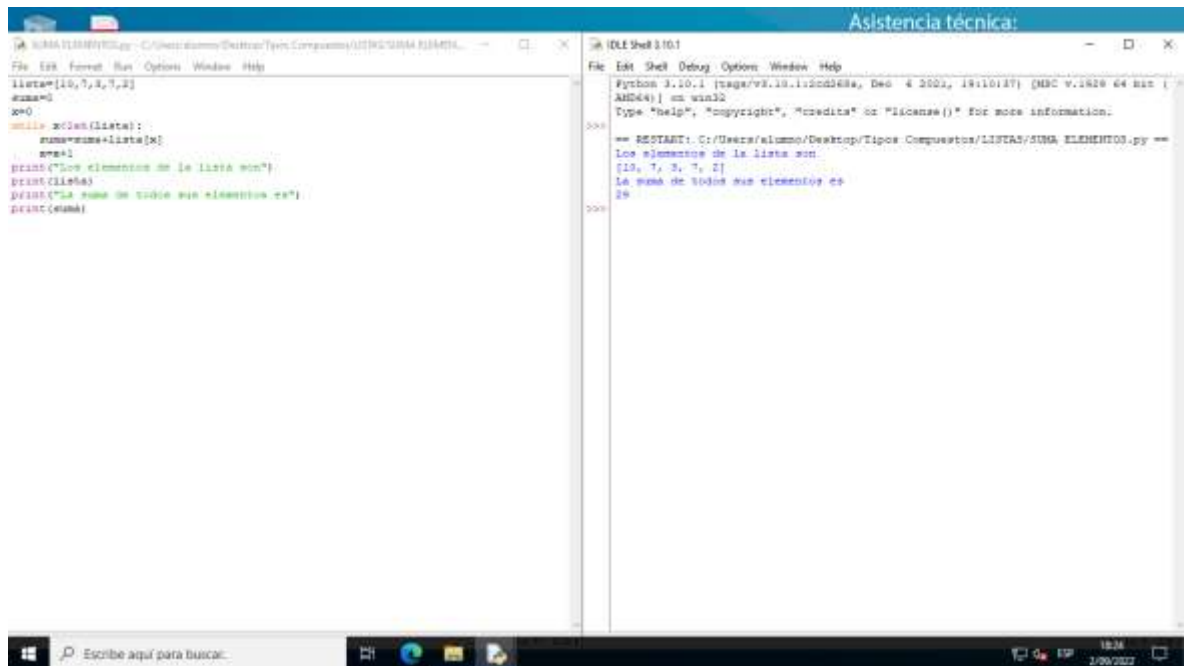
```

C:\Users\alumno\Desktop\Tipos Computacionales\LISTAS\MENOS DE LA LISTA.py
File Edit Format Run Options Window Help
#Definir una lista por asignación que almacene los nombres de los primeros cuatro
nombres["marcelo", "fernando", "marcelo", "marcelo"]
print(nombres[0]) # se muestra el valor
print(nombres[3]) # se muestra el índice

Asistencia técnica:
Python 3.10.1 (tags/v3.10.1:2002000, Dec 6 2021, 19:10:07) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/alumno/Desktop/Tipos Computacionales/LISTAS/MENOS DE LA LISTA.py =====
marcelo
marcelo
>>>

```





```
Python 3.10.1 (tags/v3.10.1:2d2068e, Dec 4 2021, 18:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> RESTART: C:/Users/usuario/Desktop/Tipos Computacion/TUPLAS/POB IN TUPLAS.py >>>
Ingresar un valor:3
Ingresar un valor:4
Ingresar un valor:5
Ingresar un valor:7
Ingresar un valor:8
Lista completa
3
4
5
7
8
El elemento mayor de la lista es 8
La suma de todos sus elementos es 27
```

```
Python 3.10.1 (tags/v3.10.1:2d2068e, Dec 4 2021, 18:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> RESTART: C:/Users/usuario/Desktop/Tipos Computacion/TUPLAS/FUNCIONES 1 TUPLAS.py >>>
Ingresar numero de dia:3
Ingresar numero de mes:4
Ingresar numero de año:2021
3/4/2021
```

```
Python 3.10.1 (tags/v3.10.1:2d248a, Dec 6 2021, 19:40:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
==== RESTART: C:/Users/alumno/Desktop/Tipos Computacion/TUPLAS/LISTAS Y TUPLAS.py ====
Imprimamos la primer tupla
(25, 12, 2014)
Imprimamos la lista que se le copio la tupla anterior
[25, 12, 2014]
Imprimamos la lista ya modificada
[31, 12, 2014]
Imprimamos la segunda tupla que se le copio la lista
(31, 12, 2014)
>>>
```

```
Python 3.10.1 (tags/v3.10.1:2d248a, Dec 6 2021, 19:40:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
==== RESTART: C:/Users/alumno/Desktop/Tipos Computacion/TUPLAS/TUPLAS.py ====
(1, 2, 3)
(25, 'Diciembre', 2014)
punto=(10, 2)
persona=('Rodriguez', 'Rafael', 43)
print(tupla)
print(fecha)
print(punto)
print(persona)
```

TUPLAS Y FUNCIONES.py - C:\Users\alumno\Desktop\Tuplas\TUPLAS Y FUNCIONES.py

File Edit Format Run Options Window Help

#3) Nombre del empleado con sueldo mayor.
#4) Cantidad de empleados con sueldo menor a 1000.

def agregar():
 empleados=[]
 while True:
 nombre=input("Nombre del empleado: ")
 sueldo=int(input("Ingrese el sueldo: "))
 empleados.append((nombre,sueldo))
 continuar=input("Continuar (s/n): ")
 if continuar.lower() != 's':
 break
 return empleados

def imprimir(empleados):
 print("Listado de los nombres de empleados y sus sueldos")
 for nombre,sueldo in empleados:
 print(nombre,sueldo)

def mayor_sueldo(empleados):
 empleado=empleados[0]
 for emp in empleados:
 if emp[1]>empleado[1]:
 empleado=emp
 print("Empleado con mayor sueldo:",empleado[0],"su sueldo es",empleado[1])

def sueldos_menor1000(empleados):
 count=0
 for empleado in empleados:
 if empleado[1]<1000:
 count=count+1
 print("Cantidad de empleados con un sueldo menor a 1000 son:",count)

bloque principal
empleados=agregar()
imprimir(empleados)
mayor_sueldo(empleados)
sueldos_menor1000(empleados)

Asistencia técnica:

File Edit Shell Debug Options Window Help

Python 3.10.1 (tags/v9.1.100100100, Dec 6 2021, 18:10:17) [AMD64] on win32
Type "help()", "copyright()", "credits()" or "license()" for more information.

-> RESTART: C:\Users\alumno\Desktop\Tuplas\TUPLAS Y FUNCIONES.py
Nombre del empleado: CESAR NAVAS
Ingrese el sueldo: 350000
Nombre del empleado: JUAN SALAZAR
Ingrese el sueldo: 2500
Nombre del empleado: RODRIGO LEMUS
Ingrese el sueldo: 899500
Nombre del empleado: TOMAS ORDOZ
Ingrese el sueldo: 50000
Nombre del empleado: ERNESTO HAJERA
Ingrese el sueldo: 300000
Listado de los nombres de empleados y sus sueldos
CESAR NAVAS 350000
JUAN SALAZAR 2500
RODRIGO LEMUS 899500
TOMAS ORDOZ 50000
ERNESTO HAJERA 300000
Empleado con mayor sueldo: RODRIGO LEMUS su sueldo es 899500
Cantidad de empleados con un sueldo menor a 1000 son: 3

->