**Ejercicio 1:**

La siguiente función te permite:

def ordenar\_asc( lista ):

ln = len(lista)

for i in range(ln):

for j in range(ln-i-1):

if( lista[j]>lista[j+1] ):

aux = lista[j]

lista[j+1] = lista[j]

lista[j+1] = aux

return lista

lista=[2,1,4,23,7]

def ordenar\_asc( lista )

print (lista)

Opciones:

a. Ordenar una lista de mayor a menor

b. Ordenar una lista de menor a mayor

c. Mantiene la lista en el orden que entra

c. Genera un Error

d. Ninguna de las otras

**Ejercicio 2:**

la siguiente función permite:

def validar():

dato=input('ingrese numero')

valido=False

while not valido:

i=0

while i<len(dato) and dato[i] in '0123456789.':

i+=1

if i==len(dato):

valido=True

else:

dato=input('ingrese nuevo número')

return float(dato)

Opciones:

a. Validar que el dato introducido sea un número entero

b. validar que el dato introducido sea una cadena formada por caracteres numéricos

c. Validar que el dato introducido sea un número real

d. Error

e. Ninguna de las otras

**Ejercicio 3:**

La siguiente función busca validar que el dato introducido sea un número real, pero el programa puede fallar.

En qué caso el programa falla:

def validar():

dato=input('ingrese numero')

valido=False

while not valido:

i=0

while i<len(dato) and dato[i] in '0123456789.':

i+=1

if i==len(dato):

valido=True

else:

dato=input('ingrese nuevo número')

return float(dato)

Opciones:

a. Cuando al inicio del dato se introduce un punto

b. Cuando el dato introducido no tiene punto

c. cuando el dato introducido tiene un punto en la última posición

d. cuando se introducen dos puntos ‘..’

e. Todas las anteriores

f. Ninguna de las otras

**Ejercicio 4:**

La siguiente función podría servir para:

def elementos(elem, secuencia):

for i in range(len(secuencia)):

if secuencia[i] == elem:

return i

Opciones:

a. Encontrar el índice del elemento en una lista

b. Encontrar el índice del elemento en una cadena

c. Encontrar el índice del elemento en un número

d. Encontrar el índice del elemento en una lista o cadena

e. Error

f. Ninguna de las otras

**Ejercicio 5:**

La siguiente función permite determinar si un elemento está en una secuencia.Qué valor devuelve si no lo encuentra:

def elementos(elem, secuencia):

for i in range(len(secuencia)):

if secuencia[i] == elem:

return i

Opciones:

a. i

b. -1

c. None

d. Error

e. Ninguna de las otras

**Ejercicio 6:**

La siguiente función debe sacar las comillas dobles (") que tiene una cadena al inicio y al final de ella. Que código debería ir dentro del if en el return ?

def sacarComillas(s):

if s[0] == '"' and s[-1] == '"':

return

else:

return s

Opciones:

a. s[:-1]

b. s[::-1]

c. s[1:-1]

d. s[1::-1]

e. Error

f. Ninguna de las otras

**Ejercicio 7:**

La siguiente función busca determinar la cantidad de días que tiene un mes según si el año es bisiesto. ¿Es correcto el código propuesto?

def cantDiasMes(mes, bisiesto):

if mes == 2 or not bisiesto:

return 28

elif mes == 2 or bisiesto:

return 29

elif mes == 4 or mes == 6 or mes == 9 or mes == 11:

return 30

else:

return 31

Opciones:

a. Código Correcto, dará la cantidad de dias según mes

b. Codigo incorrecto, no dara la cantidad de dias del los meses impares si el año es bisiesto

c. Codigo incorrecto, no dara la cantidad de dias del mes de febrero correcto si el año es bisiesto

d. Error

e. Ninguna de las otras

**Ejercicio 8:**

La siguiente función busca abrir un archivo existente para lectura.Suponga que en la dirección dada está el archivo a abrir.

¿Que imprime la función?

def leerarchivoread(file):

try:

archivo=open(file,'w')

data=archivo.read()

print("Ingrese")

archivo.close()

except:

print("El archivo no existe o no esta ubicado donde lo estas buscando")

Opciones:

a. Ingrese

b. El archivo no existe o no esta ubicado donde lo estas buscando

c. No imprime nada

d. Error

e. Ninguna de las otras

**Ejercicio 9:**

La siguiente función copia la información de una lista en un archivo csv:

def escribirarchivo(file,listanombres,listalegajos,listanotas):

with open(file,'w',newline='',encoding='utf8') as archivo:

escritor=csv.writer(archivo)

for i in range(len(listanombres)):

escritor.writerow([listanombres[i],listalegajos[i],listanotas[i]])

archivo.close()

La función close debe estar obligatoriamente porque:

Opciones:

a. Evita la pérdida de la información.

b. Evita que el archivo sea hackeado

c. Evita que me bajen 0.5 en el parcial

d. Evita que se me corrompa el archivo.

e. No se necesita en este caso

f. Ninguna de las otras

**Ejercicio 10:**

Dado el siguiente programa indicar cual es la salida esperada

def func(s):

res = ''

for i in range(-1,len(s)-1):

if s[i] in 'aeiou' :

i += 1

res += s[i]

return res

input='asdetqetyaaiop'

print(func(input))

Opciones:

a. No imprime nada

b. sttaiop

c. asdetqetyaaiop

d. steqtaaiop

e. Ninguna de las otras

**Ejercicio 11:**.

def leer(archivo):

try:po

with open(archivo, 'r', encoding = 'utf-8') as a:

encabezado = a.readline()[:-1]

except FileNotFoundError:

print('Archivo inexistente.')

return -1

Si el archivo a.csv contiene la siguiente información:

NOMBRE,EDAD

Fede, 46

Pedro, 20

Dani, 27

Indicar la salida de la siguiente ejecución:

print( leer( 'a.csv' )\*2 )

Opciones:

a. NOMBRE,EDAD

NOMBRE,EDAD

b. NOMBRE,EDADNOMBRE,EDAD

c. Fede, 46Pedro, 20Dani, 27

d. Ninguna de las otras

**Ejercicio 12:**.

Dada la siguiente función de ordenamiento indicar:

¿Qué hace la función?:

def ordenar\_asc( lista1, lista2 ):

ln = len(lista1)

for i in range(ln):

for j in range(ln-i-1):

if( lista1[j]>lista1[j+1] ):

aux = lista1[j]

lista1[j] = lista1[j+1]

lista1[j+1] = aux

aux = lista2[j]

lista2[j] = lista2[j+1]

lista2[j+1] = aux

l1 = [1,25,3,4,55]

l2 = [ 54,2,-1,3,4]

ordenar\_asc(l1,l2)

Opciones:

a. ordena la lista1 de menor a mayor, no cambia lista2

b. ordena ambas listas en función de la lista2

c. ordena la lista2 de menor a mayor, no cambia lista1

b. ordena ambas listas en función de la lista1

e. No ordeno por que falta el return

f. Ninguna de las otras

**Ejercicio 13:**.

Dado el siguiente programa, indicar qué opción del menú termina el programa?

def menuInformes():

print('''

Seleccione una opción:

1. Ranking ventas por proveedor

2. Promedio diferencia entre fecha de venta y fecha de vencimiento

3. Productos vencidos

4. Fin

''')

opcion = -1

while opcion == 1 or opcion == 2 or opcion == 3 or opcion:

menuInformes()

opcion=int(input())

print('Fin')

Opciones:

a. 2

b. 0

c. 3

d. 1

e. 5 o mayores

f. 4

g. No entra al ciclo