

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	
TEL:	DOCENTE (nombre y apellido):
AULA:	

Duración del examen: 1:20h. Completar con **letra clara, mayúscula e imprenta**. El examen consta de 11 preguntas de opción múltiple. Cada pregunta tiene una y sólo una respuesta correcta.

Las respuestas deben completarse con una X en la siguiente matriz:

Opción	EJ. 1	EJ. 2	EJ. 3	EJ. 4	EJ. 5	EJ. 6	EJ. 7	EJ. 8	EJ. 9	EJ. 10	EJ. 11
1											
2											
3											
4											

¡ATENCIÓN! Las respuestas sólo se considerarán válidas si se encuentran en la matriz. De haber diferencias entre la opción seleccionada en el ejercicio y en la matriz, se considerará como válida la de la matriz.

Ejercicio 0101 - 1 punto			
¿Qué contiene <i>b</i> ?			
<pre>def organiza(n): org=2 if n>=100 and n<1000: org=3 elif n>=1000: org=4 return org a=[1,15,150,1500,2,20,200,2000] b=list(map(organiza,a))</pre>			
1	[2, 2, 2, 2, 4, 4, 4, 4]		1
2	[]		2
3	[1, 2, 3, 4, 5, 6, 7, 8]		3
4	[2, 2, 3, 4, 2, 2, 3, 4]	X	4

Ejercicio 0201 - 1 punto			
¿ Qué versión de la función ingreso() valida correctamente los datos de acceso de usuarios? Deben coincidir usuario y clave			
<pre># users contiene usuario:clave users={'jmiguel':'X25fc230','irojas':'Irma1211', 'jsal':'joacoS01','lledesma':'LEDElun9'} while not ingreso(users): print('Datos de Acceso Erróneos')</pre>			
1	<pre>def ingreso(usr): u=input('Usuario: ') cl=input('Clave: ') if u in usr: return True else: return False</pre>		1
2	<pre>def ingreso(usr): u=input('Usuario: ') cl=input('Clave: ') if u in usr: if usr[u]==cl: return True return False</pre>	X	2
3	<pre>def ingreso(usr): u=input('Usuario: ') cl=input('Clave: ') if u in usr: if cl in usr: return True</pre>		3
4	<pre>def ingreso(usr): u=input('Usuario: ') cl=input('Clave: ') if cl in usr: if usr[u]==cl: return True return False</pre>		4

Ejercicio 0301 - 1 punto			
¿Qué versión de la función entPositivo() valida correctamente el ingreso de un número natural (entero positivo)?			
Nota: Se espera que la función no termine hasta obtener y devolver un valor correcto			
<pre>num=entPositivo('Ingresá un número natural: ')</pre>			
1	<pre>def entPositivo(cartel): pedir=True while pedir: try: n=int(input(cartel)) if n>0: pedir=False else: print('Tiene que ser mayor que 0') except ValueError: print('Número entero') return n</pre>	X	1
2	<pre>def entPositivo(cartel): pedir=True while pedir: try: n=int(input(cartel)) except ValueError: print('Número entero') pedir=False return n</pre>		2
3	<pre>def entPositivo(cartel): for i in range(3): try: n=int(input(cartel)) except ValueError: print('Número entero') return n</pre>		3
4	<pre>def entPositivo(cartel): try: n=int(input(cartel)) return n except ValueError: print('Número entero')</pre>		4

Ejercicio 0401 - 1 punto			
<p>¿Cuál versión de la función cuenta() devuelve la cantidad de nombres que comienzan con A en el archivo nombres, que tiene 3 líneas con nombres separados por -?</p> <p>Contenido de nombres.txt:</p> <p>Andrea-LAURA-MÁXIMO-Ileana jorgelina-Silvana-amalia-alberto-andrés ALMA-ELENA</p> <p>cuenta() debe devolver 5</p> <pre>def cuenta(arch,letras): - - datos=open('nombres.txt',encoding='utf-8') inicia='a','á' cantidad=cuenta(datos,inicia) datos.close() print(cantidad,'nombres inician con',inicia[0].upper())</pre>			
1	<pre>def cuenta(arch,letras): cant=0 for i in range(3): linea=arch.readline() if linea[0] in letras: cant+=1 return cant</pre>		1
2	<pre>def cuenta(arch,letras): cant=0 lineas=arch.readlines() for i in range(len(lineas)): lineas[i]=lineas[i].strip('\n').split('-') for nom in lineas[i]: if nom[0].lower() in letras: cant+=1 return cant</pre>	X	2
3	<pre>def cuenta(arch,letras): cant=0 lineas=arch.readlines() for i in range(len(lineas)): for nom in lineas[i]: if nom[0].lower() in letras: cant+=1 return cant</pre>		3
4	<pre>def cuenta(arch,letras): for i in range(3): linea=arch.readline() linea=linea.split('-') cant=linea.count(letras[0])+linea.count(letras[1]) return cant</pre>		4

Ejercicio 0501 - 1 punto																															
<p>Dado el siguiente DataFrame <i>recaudacion</i>:</p> <table border="1" data-bbox="138 1908 722 2271"> <thead> <tr> <th></th> <th>cajero</th> <th>caja</th> <th>total</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>juan</td> <td>3</td> <td>104000.77</td> </tr> <tr> <td>1</td> <td>ana</td> <td>1</td> <td>256000.00</td> </tr> <tr> <td>2</td> <td>mario</td> <td>5</td> <td>99025.40</td> </tr> <tr> <td>3</td> <td>juan</td> <td>1</td> <td>112000.00</td> </tr> <tr> <td>4</td> <td>ana</td> <td>2</td> <td>77090.00</td> </tr> <tr> <td>5</td> <td>mario</td> <td>3</td> <td>86450.00</td> </tr> </tbody> </table> <p>Que contiene 6 filas y 3 columnas: nombre del cajero (cajero), número de la caja en que cobró (caja) y total cobrado en esa ocasión (total).</p> <p>¿Qué instrucción produce la siguiente salida?</p> <pre>cajero ana 333090.00 juan 216000.77 mario 185475.40</pre>					cajero	caja	total	0	juan	3	104000.77	1	ana	1	256000.00	2	mario	5	99025.40	3	juan	1	112000.00	4	ana	2	77090.00	5	mario	3	86450.00
	cajero	caja	total																												
0	juan	3	104000.77																												
1	ana	1	256000.00																												
2	mario	5	99025.40																												
3	juan	1	112000.00																												
4	ana	2	77090.00																												
5	mario	3	86450.00																												

1	recaudacion.loc[:,['total']]		1
2	recaudacion['cajero'].value_counts()		2
3	recaudacion.groupby(recaudacion['cajero'])['total'].sum()	X	3
4	recaudacion.groupby(recaudacion['cajero']).min()		4

Ejercicio 0601 - 1 punto

Dado el siguiente programa:

```
def obtieneNom(t):
    return t.upper()

nombres=['ana López','emiliano SAL','LORENA ana báunes',
        'analía Soto','ángelea FALCÓN',
        'Luciana analía pÉrez','Ana María Giménez']
```

resultado= . . .

```
for nom in resultado:
    print(nom)
```

Que produce la siguiente salida:

```
ANA LÓPEZ
EMILIANO SAL
LORENA ANA BÁUNES
ANALÍA SOTO
ÁNGELEA FALCÓN
LUCIANA ANALÍA PÉREZ
ANA MARÍA GIMÉNEZ
>>>
```

Qué instrucción debería ir en los puntos suspensivos?

Nota: El argumento *key* permite pasarle a la función un criterio alternativo de comparación entre los elementos de la estructura. En este caso se comparan las versiones de los nombres en mayúsculas.

1.	max(nombres, key=obtieneNom)		1
2.	list(map(obtieneNom, nombres))	X	2
3.	sorted(nombres, key=obtieneNom)		3
4.	list(filter(obtieneNom, nombres))		4

Ejercicio 0701 - 1 punto

Dado el siguiente programa:

```
print('Ingresá números enteros, 0 para terminar')
num=1
i=1
lista=[]
while num!=0:
    num=input(str(i)+' : ')
    try:
        num=int(num)
        lista.append(num)
    except ValueError:
        lista.append(0)
    i+=1
while 0 in lista:
    lista.remove(0)
```

Y los siguientes ingresos:

```
1: 1-2
2: 12
3: me equivoqué
4: 3.56
5: -12
6: 0
```

¿Qué contenido tendrá *lista* al finalizar?

1	12		1
2	[12, -12]	X	2
3	[0, 12, 0, 0, -12, 0]		3
4	[3, 12, 3.56, -12]		4

Ejercicio 0801 - 1 punto



¿Cuál de las siguientes líneas de código NO SE CORRESPONDE con la figura?

1.	<code>ax.set_xlim(0, 7)</code>	X	1
2.	<code>ax.grid()</code>		2
3.	<code>ax.legend()</code>		3
4.	<code>ax.set_title("Gráfico de ventas")</code>		4

Ejercicio 0901 - 1 punto

Luego de ejecutar el siguiente fragmento de código:

```
X = [1,2,3,4,5]
Y = [1,2,3,4,5]

fig, ax = plt.subplots(figsize=(6, 6), dpi=80)
ax.plot(X, Y)
ax.set_xlim(X[-1], X[0])
ax.set_ylim(0, Y[-1])
ax.set_title('Título 1')
ax.set_title('Título 2')
plt.show()
```

¿Cuál de las siguientes opciones describe correctamente el gráfico que se ilustrara?

1.	En el gráfico se va a ver una línea que va de abajo a la izquierda hacia arriba a la derecha con título "Título 2"		1
2.	En el gráfico se va a ver una línea que va de arriba a la izquierda hacia abajo a la derecha con título "Título 2"	X	2
3.	En el gráfico se va a ver una línea que va de abajo a la izquierda hacia arriba a la derecha con dos títulos solapados: "Título 1" y "Título 2"		3
4.	Ninguna de las opciones describe lo que ocurre al ejecutar el código		4

Ejercicio 1001 - 2 puntos

¿Cómo queda el archivo `vtasSemestre.txt` luego de ejecutar el siguiente programa?

```
def leer(arch):
    vtasMes={1:0,2:0,3:0,4:0,5:0,6:0}
    ventas=open(arch,encoding='utf-8')
    detalle=ventas.readlines()
    ventas.close()
    for i in range(len(detalle)):
        detalle[i]=detalle[i].strip().split(',')
        mes=int(detalle[i][0])
        vta=float(detalle[i][1])
        vtasMes[mes]=vtasMes[mes]+vta
    return vtasMes

def guarda(dicci,arch,meses):
    datos=open(arch,'w',encoding='utf-8')
    for mes in dicci:
        lin=meses[mes-1]+' '+str(dicci[mes])+'\n'
        datos.write(lin)
    datos.close()

meses= 'enero','febrero','marzo','abril','mayo','junio'
ventas=leer('detalleVtas.txt')
guarda(ventas,'vtasSemestre.txt',meses)
```

Nota: El contenido de `detalleVtas.txt` es el siguiente:

- 1,25
- 2,33
- 1,33.5
- 3,20
- 4,10
- 1,11

Por cada línea viene el número de mes y el valor de una venta separados por coma

1.	0:11 1:33.0 2:20.0 3:10.0		1
2.	enero:69.5 febrero:33.0 marzo:20.0 abril:10.0 mayo:0 junio:0	X	2
3.	ENERO:69.5		3
4.	enero,69.5,febrero,33.0,marzo,20.0,abril,10.0,mayo,junio		4

Ejercicio 1101 - 2 puntos

Dado el siguiente DataFrame *menu*:

	nombre	principal	caloría
lunes	tallarines	hidratos	1500
martes	milanesas	proteínas	1800
miércoles	tarta de choclo	hidratos	1000
jueves	lenguado	proteínas	950

Que contiene 4 filas (el día de la semana es el índice de la fila) y 3 columnas: nombre del plato (nombre), grupo nutricional principal (principal) y calorías por porción (calorías).

¿Qué muestra como resultado la siguiente instrucción?

```
menu[(menu['principal']=='proteínas') & (menu['calorías']<1500)]
```

1	nombre calorías Lunes tallarines 1500 Jueves lenguado 950		1
2	nombre principal calorías Miércoles tarta de choclo hidratos 1000		2
3	nombre principal calorías Martes milanesas proteínas 1800 Jueves lenguado proteínas 950		3
4	nombre principal calorías Jueves lenguado proteínas 950	X	4