

Ejercicio 1 – Tema 3		1 Pto
Cuál de los siguientes diccionarios tiene todas sus claves no numéricas (no son tipo int, float ni complex)?		
A	{1:2, 2:'febrero', True:False, '1':'enero'}	A
B	{True:'V', '1':'uno', '4':'cuatro', False:'F'}	X B
C	{'3':0, 2:66, 5:3.45, (1,):[-2,5]}	C
D	{('3','5'):44, 2:('1','0')}	D

Ejercicio 2 – Tema 3		1 Pto												
¿Cuál de las siguientes instrucciones debería usar (junto a otras) para generar un gráfico como el siguiente con matplotlib? Los datos a representar son: <pre>precio = [3400, 1915, 2000, 7200, 1270] productos= ['chocolate', 'manteca', 'azúcar', 'huevo', 'harina']</pre>														
<div style="text-align: center;"> <p>Precio Unitario Productos de Despensa</p> <table border="1"> <caption>Data for Precio Unitario Productos de Despensa</caption> <thead> <tr> <th>Producto</th> <th>Precio en \$</th> </tr> </thead> <tbody> <tr> <td>harina</td> <td>1200</td> </tr> <tr> <td>huevo</td> <td>7200</td> </tr> <tr> <td>azúcar</td> <td>2000</td> </tr> <tr> <td>manteca</td> <td>1915</td> </tr> <tr> <td>chocolate</td> <td>3400</td> </tr> </tbody> </table> </div>			Producto	Precio en \$	harina	1200	huevo	7200	azúcar	2000	manteca	1915	chocolate	3400
Producto	Precio en \$													
harina	1200													
huevo	7200													
azúcar	2000													
manteca	1915													
chocolate	3400													
A	ax.barh(productos,precio)	X A												
B	plt.grid(precio)	B												
C	ax.bar(precio, productos)	C												
D	ax.scatter(precio)	D												

Ejercicio 3 – Tema 3		1 Pto
Dado el Dataframe resfrios :		
<pre> mes humedad cantidad 0 enero 60 2 1 febrero 75 23 2 marzo 79 32 3 abril 68 31 4 mayo 62 45 5 junio 55 61 </pre>		
¿Cuál de las instrucciones debería usar para producir la siguiente salida?		
<pre> cantidad mes 0 2 enero 1 23 febrero 2 32 marzo 3 31 abril 4 45 mayo 5 61 junio </pre>		
A	resfrios['humedad'].mean()	A
B	resfrios[['cantidad', 'mes']]	X B
C	resfrios.loc[resfrios.index[[1,3,5]], ['humedad', 'cantidad']]	C
D	resfrios[resfrios['cantidad']>=32]	D

Ejercicio 4 – Tema 3		1 Pto	
<p>Dado el archivo carreras.txt, que contiene alumnos anotados en distintas carreras:</p> <pre> Quimica Analia,Ricardes,2021 Juliana,Ortesano,2023 Pedro,Fuentes,2023 Industrial Jose,Agustoni,2021 Ana,Mauner,2022 Andrea,Carragno,2022 Civil Marta,Leiva,2021 </pre> <p>¿Con cuál conjunto de modos de apertura funcionará adecuadamente el siguiente programa?</p> <pre> grupos=open('carreras.txt', ...) #modo 1? carreras=open('listado-carreras.txt', ...) #modo 2? alumnos=open('listado-alumnos.txt', ...) #modo 3? carr=grupos.readline() carreras.write(carr) for lin in grupos: if ',' in lin: alumnos.write(lin) else: carreras.write(lin) alumnos.close() carreras.close() grupos.close() </pre> <p>Al finalizar quedarán 2 archivos adicionales: listado-alumnos.txt, que contendrá la lista total de alumnos que hay en carreras.txt; y listado-carreras.txt, que contendrá la lista total de carreras que figuran en carreras.txt</p>			
A	<pre> modo 1 'r' modo 2 'w' modo 3 'w' </pre>	X	A
B	<pre> modo 1 'a' modo 2 'w' modo 3 'r' </pre>		B
C	<pre> modo 1 'r' modo 2 'r' modo 3 'r+' </pre>		C
D	<pre> modo 1 'w' modo 2 'w' modo 3 'w' </pre>		D

Ejercicio 5 – Tema 3		1 Pto	
<p>¿Cuál programa realiza un ingreso validado de un múltiplo de 10?</p> <ul style="list-style-type: none"> ✓ No se debe aceptar un número que no sea múltiplo de 10 ✓ El programa no debe dar fallo si se ingresa un dato que no sea entero ✓ El programa debe insistir en la solicitud hasta obtener lo esperado 			
A	<pre>ok=False while not ok: try: num=int(input('Ingresá un múltiplo de 10: ')) if num>0 and num%10==0: ok=True else: print('No es múltiplo de 10') except ValueError: print('Debe ser número entero') print(num)</pre>	X	A
B	<pre>pide=False while pide: try: num=int(input('Ingresá un múltiplo de 10: ')) if num<0 or num%10!=0: pide=False else: print('No es múltiplo de 10') except ValueError: print('Debe ser número entero') print(num)</pre>		B
C	<pre>for i in range(3): num=int(input('Ingresá un múltiplo de 10: ')) if num<0 or num%10!=0: print('No es múltiplo de 10') print(num)</pre>		C
D	<pre>try: num=int(input('Ingresá un múltiplo de 10: ')) except: print('No es un número') if num>0 and num%10==0: print(num)</pre>		D

Ejercicio 6 – Tema 3			2 Ptos																												
Dado el Dataframe <i>resfrios</i> :																															
<table border="1"> <thead> <tr> <th></th> <th>mes</th> <th>humedad</th> <th>cantidad</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>enero</td> <td>60</td> <td>2</td> </tr> <tr> <td>1</td> <td>febrero</td> <td>75</td> <td>23</td> </tr> <tr> <td>2</td> <td>marzo</td> <td>79</td> <td>32</td> </tr> <tr> <td>3</td> <td>abril</td> <td>68</td> <td>31</td> </tr> <tr> <td>4</td> <td>mayo</td> <td>60</td> <td>45</td> </tr> <tr> <td>5</td> <td>junio</td> <td>60</td> <td>61</td> </tr> </tbody> </table>					mes	humedad	cantidad	0	enero	60	2	1	febrero	75	23	2	marzo	79	32	3	abril	68	31	4	mayo	60	45	5	junio	60	61
	mes	humedad	cantidad																												
0	enero	60	2																												
1	febrero	75	23																												
2	marzo	79	32																												
3	abril	68	31																												
4	mayo	60	45																												
5	junio	60	61																												
¿Qué salida produce la siguiente instrucción?																															
<pre>print(resfrios.sort_values(by=['humedad'], ascending=[True]))</pre>																															
A	<table border="1"> <thead> <tr> <th></th> <th>mes</th> <th>humedad</th> <th>cantidad</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>enero</td> <td>60</td> <td>2</td> </tr> <tr> <td>4</td> <td>mayo</td> <td>60</td> <td>45</td> </tr> <tr> <td>5</td> <td>junio</td> <td>60</td> <td>61</td> </tr> <tr> <td>3</td> <td>abril</td> <td>68</td> <td>31</td> </tr> <tr> <td>1</td> <td>febrero</td> <td>75</td> <td>23</td> </tr> <tr> <td>2</td> <td>marzo</td> <td>79</td> <td>32</td> </tr> </tbody> </table>		mes	humedad	cantidad	0	enero	60	2	4	mayo	60	45	5	junio	60	61	3	abril	68	31	1	febrero	75	23	2	marzo	79	32	X	A
	mes	humedad	cantidad																												
0	enero	60	2																												
4	mayo	60	45																												
5	junio	60	61																												
3	abril	68	31																												
1	febrero	75	23																												
2	marzo	79	32																												
B	<table border="1"> <thead> <tr> <th></th> <th>cantidad</th> <th>mes</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2</td> <td>enero</td> </tr> <tr> <td>1</td> <td>23</td> <td>febrero</td> </tr> <tr> <td>2</td> <td>32</td> <td>marzo</td> </tr> <tr> <td>3</td> <td>31</td> <td>abril</td> </tr> <tr> <td>4</td> <td>45</td> <td>mayo</td> </tr> <tr> <td>5</td> <td>61</td> <td>junio</td> </tr> </tbody> </table>		cantidad	mes	0	2	enero	1	23	febrero	2	32	marzo	3	31	abril	4	45	mayo	5	61	junio		B							
	cantidad	mes																													
0	2	enero																													
1	23	febrero																													
2	32	marzo																													
3	31	abril																													
4	45	mayo																													
5	61	junio																													
C	<table border="1"> <thead> <tr> <th>humedad</th> <th></th> </tr> </thead> <tbody> <tr> <td>60</td> <td>108</td> </tr> <tr> <td>68</td> <td>31</td> </tr> <tr> <td>75</td> <td>23</td> </tr> <tr> <td>79</td> <td>32</td> </tr> </tbody> </table>	humedad		60	108	68	31	75	23	79	32		C																		
humedad																															
60	108																														
68	31																														
75	23																														
79	32																														
D	<table border="1"> <thead> <tr> <th>humedad</th> <th></th> </tr> </thead> <tbody> <tr> <td>60</td> <td>3</td> </tr> <tr> <td>75</td> <td>1</td> </tr> <tr> <td>79</td> <td>1</td> </tr> <tr> <td>68</td> <td>1</td> </tr> </tbody> </table>	humedad		60	3	75	1	79	1	68	1		D																		
humedad																															
60	3																														
75	1																														
79	1																														
68	1																														

Ejercicio 7 – Tema 3		2 Ptos
<p>Dado el siguiente programa:</p> <pre>def calc1(pal): voc=0 for letra in pal.lower(): if letra in 'ááééííoóúúü': voc+=1 return voc>2 def calc2(pal): conso=0 for letra in pal.lower(): if letra not in 'ááééííoóúúü': conso+=1 return conso def calc3(pal): return len(pal) palabras=['casa', 'auto', 'banco', 'cartel', 'semáforo'] # línea de código faltante print(nuevaLista)</pre> <p>¿Cuál debería ser la línea de código faltante para que la salida sea la siguiente? [2, 1, 3, 4, 4]</p>		
A	<code>nuevaLista=list(map(calc3,palabras))</code>	A
B	<code>nuevaLista.insert(map(calc1,palabras))</code>	B
C	<code>nuevaLista=list(map(calc2,palabras))</code>	X C
D	<code>nuevaLista.sort(map(edital,palabras))</code>	D

Ejercicio 8 – Tema 3		2 Ptos	
<p>Dado el siguiente programa:</p> <pre>sucursales=... #?? vtasSuc=[17800,250010,67509,148000,333000,139000,72900] print('Ventas Mayo por Sucursales') for i in range(len(vtasSuc)): if i in sucursales: print(sucursales[i][1],',',sucursales[i][0], ':\$',vtasSuc[i])</pre> <p>¿Cuál de las opciones de la estructura sucursales funcionará correctamente para producir la siguiente salida?</p> <pre>Ventas Mayo por Sucursales CABA , Flores :\$ 17800 CABA , Devoto :\$ 250010 CABA , Once :\$ 67509 CBA , Urca :\$ 333000 CBA , Rogelio Martínez :\$ 72900</pre>			
A	<pre>sucursales={0:['Flores','CABA'],2:['Once','CABA'], 4:['Urca','CBA'],6:['Rogelio Martínez','CBA'], 1:['Devoto','CABA']}</pre>	X	A
B	<pre>sucursales=[['Flores','CABA'],['Once','CABA'], ['Urca','CBA'],['Rogelio Martínez','CBA']]</pre>		B
C	<pre>sucursales={'suc0':['Flores','CABA'], 'suc2':['Once','CABA'], 'suc4':['Urca','CBA'], 'suc6':['Rogelio Martínez','CBA'], 'suc1':['Devoto','CABA']}</pre>		C
D	<pre>sucursales={'CABA':['Flores','suc0'], 'CABA':['Once','suc2'], 'CBA':['Urca','suc4'], 'CBA':['Rogelio Martínez','suc6'], 'CABA':['Devoto','suc1']}</pre>		D

Ejercicio 9 – Tema 3		2 Ptos	
<p>¿Cuál es la salida del siguiente programa?</p> <pre>def edita (txt): return txt.upper() def patron(txt): return ('MB' in txt) or ('LT' in txt) or ('LD' in txt) ciudades=['anillaco', 'salta','casilda','fiambalá'] salida=list(map(edita,ciudades)) mostrar=list(filter(patron,salida)) print(mostrar)</pre>			
A	[]		A
B	['SALTA', 'CASILDA', 'FIAMBALÁ']	X	B
C	['ANILLACO']		C
D	['anillaco', 'san javier', 'la dársena', 'fiambalá']		D