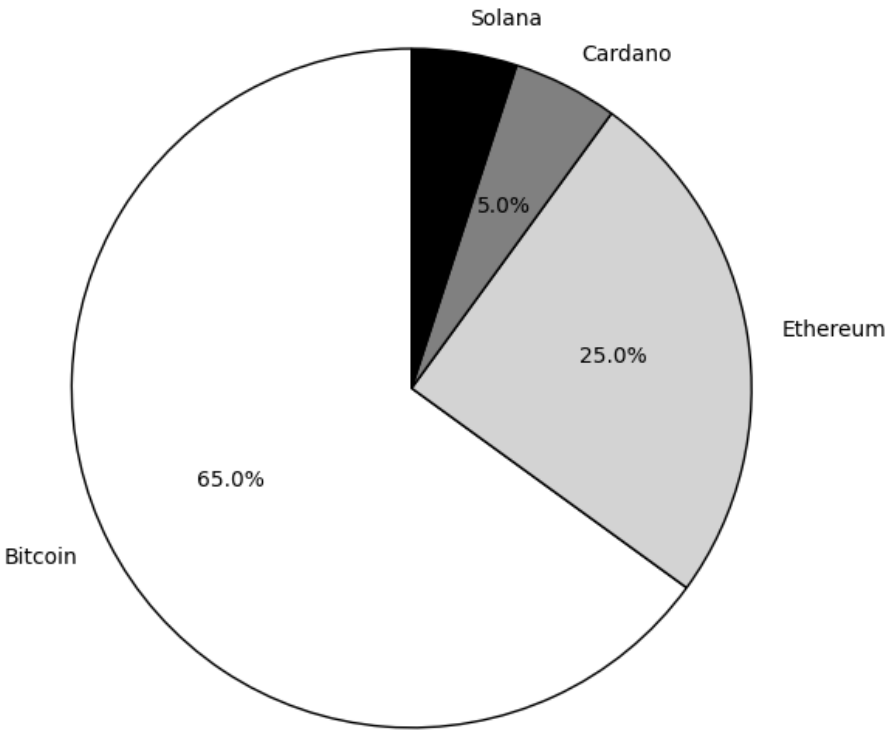


Ejercicio 1 – Tema 1			1 Pto
Cuál de los siguientes diccionarios tiene tipos diferentes de claves (tipo int, float, str, tuple, bool ...)?			
A	{ '1':2, '23':5, 0:13 }	X	A
B	{ 1:'uno', 2:'dos', 3:'V', 0:'F' }		B
C	{ '0': '3', '2':66, '5':3.45, '1': [-2,5] }		C
D	{ ('1', '0'):2, ('3', '5'):44 }		D

Ejercicio 2 – Tema 1			1 Pto
¿Cuál de las siguientes instrucciones debería usar (junto a otras) para generar un gráfico como el siguiente con matplotlib?			
Participación en Ventas - Escala de Grises 			
A	<code>plt.bar(criptos, ventas, color=colores, edgecolor='black')</code>		A
B	<code>plt.title('SOL')</code>		B
C	<code>plt.ylabel('Millones de USD')</code>		C
D	<code>plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90, wedgeprops={'edgecolor': 'black'})</code>	X	D

Ejercicio 3 – Tema 1			1 Pto
Dado el Dataframe <i>clientes</i> :			
<pre>apellido edad ciudad provincia 0 Antón 50.0 Chajarí Entre Ríos 1 Khunter NaN Concordia Entre Ríos 2 Ortúzar 44.0 La Carlota Córdoba 3 Gómez 23.0 Chajarí Entre Ríos 4 Arce 35.0 La Paz Entre Ríos 5 Alves 31.0 Salsipuedes Córdoba 6 Alves 41.0 None None 7 Díaz 72.0 Tafí Viejo Tucumán</pre>			
¿Cuál de las siguientes instrucciones debería usar para producir la siguiente salida?			
<pre>ciudad Chajarí 2 Concordia 1 La Carlota 1 La Paz 1 Salsipuedes 1 Tafí Viejo 1</pre>			
A	<code>clientes.head(2)</code>		A
B	<code>clientes.sort_values(by=['provincia','ciudad'], ascending=[True,False])</code>		B
C	<code>clientes.groupby('provincia')['edad'].min()</code>		C
D	<code>clientes['ciudad'].value_counts()</code>	X	D

Ejercicio 4 – Tema 1			1 Pto
Dado los archivos <i>vtas-suc1.txt</i> y <i>vtas-suc2.txt</i> que contienen las ventas mensuales del primer semestre de cada sucursal:			
vtas-suc1.txt ene,69.5 feb,33.0 mar,20.0 abr,10.0 may,0 jun,0 vtas-suc2.txt ene,19.5 feb,43.0 mar,0.0 abr,110.0 may,11 jun,66			
¿Cuál es la sentencia que falta en el programa? Al finalizar quedará 1 archivo adicional: vtas-semester.txt , que contendrá una línea con el total de las ventas del semestre de las dos sucursales			
<pre>total=0 suc=open('vtas-suc1.txt') todas=suc.readlines() suc.close() for elem in todas: lin=elem.strip('\n') lin=lin.split(',') total+=float(lin[1]) suc=open('vtas-suc2.txt') todas=suc.readlines() suc.close() for elem in todas: lin=elem.strip('\n') lin=lin.split(',') total+=float(lin[1]) linea='Total ventas semestre sucursal 1 y 2: \$'+str(total) semestre=open('vtas-semester.txt','w') ... # línea faltante semestre.close()</pre>			
A	<code>semestre=readlines()</code>		A
B	<code>semestre.write(linea)</code>	X	B
C	<code>semestre=open('vtas-semester.txt')</code>		C
D	<code>semestre=close(write)</code>		D

Ejercicio 5 – Tema 1		1 Pto	
¿Cuál programa logra rellenar todos los elementos de la lista suma sin que se produzca fallo cuando se acaban los elementos de la lista lis2 ? La salida correcta debe ser: [11, 22, -2, 2, 1, 1]			
Nota: El operador * aplicado a una lista es el operador repetición Ej: [1,2]*3 -> [1,2,1,2,1,2]			
A	<pre>lis1=[10,-3,6,2,1,1] lis2=[1,25,-8] suma=[] for i in range(len(lis1)): total=lis1[i]+lis2[i] try: suma.append(total) except IndexError: print('No hay más lugar') print(suma)</pre>		A
B	<pre>lis1=[10,-3,6,2,1,1] lis2=[1,25,-8] suma=[0]*len(lis1) for i in range(len(lis1)): try: suma[i]=lis1[i]+lis2[i] except IndexError: suma[i]=lis1[i] print(suma)</pre>	X	B
C	<pre>lis1=[10,-3,6,2,1,1] lis2=[1,25,-8] suma=[0]*len(lis1) for i in range(len(lis1)): total=lis1[i]+lis2[i] try: suma[i]=total except IndexError: suma[i]=0 print(suma)</pre>		C
D	<pre>lis1=[10,-3,6,2,1,1] lis2=[1,25,-8] try: suma=[0]*len(lis1) except ValueError: suma=lis1+lis2 for i in range(len(lis1)): suma[i]=lis1[i]+lis2[i] print(suma)</pre>		D

Ejercicio 6 – Tema 1				2 Ptos	
Dado el Dataframe <i>clientes</i> :					
	apellido	edad	ciudad	provincia	
0	Antón	50.0	Chajarí	Entre Ríos	
1	Khunter	NaN	Concordia	Entre Ríos	
2	Ortúzar	44.0	La Carlota	Córdoba	
3	Gómez	23.0	Chajarí	Entre Ríos	
4	Arce	35.0	La Paz	Entre Ríos	
5	Alves	31.0	Salsipuedes	Córdoba	
6	Alves	41.0	None	None	
7	Díaz	72.0	Tafí Viejo	Tucumán	
¿Qué salida produce la siguiente instrucción?					
<pre>print(clientes.sort_values(by=['provincia','ciudad'], ascending=[True,False]))</pre>					
A	provincia		ciudad	apellido	
	0	Entre Ríos	Chajarí	Antón	
	1	Entre Ríos	Concordia	Khunter	
	2	Córdoba	La Carlota	Ortúzar	
	3	Entre Ríos	Chajarí	Gómez	
	4	Entre Ríos	La Paz	Arce	
	5	Córdoba	Salsipuedes	Alves	
	6	None	None	Alves	
7	Tucumán	Tafí Viejo	Díaz		
B	apellido	edad	ciudad	provincia	
	5	Alves	31.0	Salsipuedes	Córdoba
	2	Ortúzar	44.0	La Carlota	Córdoba
	4	Arce	35.0	La Paz	Entre Ríos
	1	Khunter	NaN	Concordia	Entre Ríos
	0	Antón	50.0	Chajarí	Entre Ríos
	3	Gómez	23.0	Chajarí	Entre Ríos
	7	Díaz	72.0	Tafí Viejo	Tucumán
C	apellido	edad	ciudad	provincia	
	5	Alves	31.0	Salsipuedes	Córdoba
	6	Alves	41.0	None	None
	7	Díaz	72.0	Tafí Viejo	Tucumán
D	apellido	edad	ciudad	provincia	
	2	Ortúzar	44.0	La Carlota	Córdoba
	3	Gómez	23.0	Chajarí	Entre Ríos
	4	Arce	35.0	La Paz	Entre Ríos
	5	Alves	31.0	Salsipuedes	Córdoba

Ejercicio 7 – Tema 1			2 Ptos
Dado el siguiente programa: <pre>def naipe1(t): return t def naipe2(tup): return tup[0]+' de '+tup[1].upper() def naipe3(tup): return list(tup) def naipe4(tup): return tup[1]+' , '+tup[0].upper() mano=[('10','trébol'),('2','diamante'), ('10','diamante'),('Jota','trébol')] # línea de código faltante for carta in detalle: print(carta)</pre> ¿Cuál debería ser la línea de código faltante para que la salida sea la siguiente? <pre>10 de TRÉBOL 2 de DIAMANTE 10 de DIAMANTE Jota de TRÉBOL</pre>			
A	detalle=list(map(naipe2,mano))	X	A
B	detalle.insert(0,edita3(mano))		B
C	detalle=list(map(naipe4,mano))		C
D	detalle=str(map(mano,naipe1))		D

Ejercicio 8 – Tema 1		2 Ptos	
Dado el siguiente programa: <pre>presentaciones=... #?? pedido=['té','café','jugo','yogurt'] print('Pedido') print(*pedido) print('Infusiones del pedido que tienen presentación en saquito:') for elem in pedido: if elem in presentaciones and 'saquito' in presentaciones[elem]: print(elem)</pre> ¿Cuál de las opciones de la estructura presentaciones funcionará correctamente para producir la siguiente salida? Pedido té café jugo yogurt Infusiones del pedido que tienen presentación en saquito: té café Nota: Un * delante de una lista en un print() saca un elemento al lado del otro, separados por espacio Ej: print(*[1,2,3]) -> 1 2 3			
A	<pre>presentaciones={'té':['hebras','saquito'], 'café':['granos','molido', 'cápsula','saquito'], 'yerba':['saquito','con palo', 'sin palo'], 'chocolatada':['tetra','polvo']}</pre>	X	A
B	<pre>presentaciones=['té','hebras','saquito','café', 'granos','molido','cápsula', 'saquito','yerba','saquito', 'con palo','sin palo', 'chocolatada','tetra','polvo']</pre>		B
C	<pre>presentaciones={1:['té','hebras','saquito'], 2:['café','granos','molido','cápsula', 'saquito'], 11:['yerba','saquito','con palo', 'sin palo'], 5:['chocolatada','tetra','polvo']}</pre>		C
D	<pre>presentaciones={'saquito':['café','té','yerba'], 'molido':['café'], 'granos':['café'], 'polvo':['chocolatada']}</pre>		D

Ejercicio 9 – Tema 1		2 Ptos	
¿Cuál es la salida del siguiente programa?			
<pre>def edita (lis): return lis[0].upper() def patron(lis): inicial=lis[1][0].lower() return inicial=='a' vecinos=[['clara','diaz'], ['MARTÍN', 'ARGAÑARAZ'], ['Silvio', 'Lusikcatz'], ['mariela', 'ruiz'], ['maría', 'alves']] datosValidos=list(filter(patron,vecinos)) salida=list(map(edita,datosValidos)) print(salida)</pre>			
A	['MARTÍN', 'MARÍA']	X	A
B	['Claro', 'Martín', 'Silvio', 'Mariela', 'María']		B
C	[]		C
D	['diaz', 'lusikcatz', 'ruiz']		D