

Ejercicio 1 – Tema 1			1 Pto
¿Con cuál grupo de datos el siguiente programas muestra por pantalla decena? <pre>..... #inicialización de variables if buscaNum in num: if buscaNum>10: print('Dos docenas o Docena y Media') elif buscaNum in range(1,4,2): print('Algunos') else: print('Un par') else: if buscaTxt in medidas: for med in medidas: if buscaTxt==med: print(med)</pre>			
A	num=(1,2,3,18,24) medidas=('docena','centena','decena','par') buscaNum=10 buscaTxt='decena'	X	A
B	num=(1,2,3,12,24) medidas=('docena','centena','decena','par') buscaNum=12 buscaTxt='decena'		B
C	num=(2,18,24) medidas=('docena','centena','decena','par') buscaNum=1 buscaTxt='uno'		C
D	num=(1,2,3,18,24) medidas=('docena','centena','decena','par') buscaNum=10 buscaTxt='par'		D

Ejercicio 2 – Tema 1			1 Pto
¿Qué muestra el siguiente programa? <pre>a=10 b=0 c=2.5 d=-1 print((c+b)*2/a+c-d)</pre>			
A	3.75		A
B	2		B
C	4.0	X	C
D	0		D

Ejercicio 3 – Tema 1		1 Pto	
¿Qué contenido tendrán al final las listas listaA , listaB y listaC en el siguiente programa?			
<pre>indices=[1,1,2,1,2,2,2,1] nombres=['lucas','Ana','inés','juan','JIMENA', 'Joaquín','mariana','rafa'] listaA=[] listaB=[] listaC=[] for i in range(len(nombres)): if indices[i]==1: listaA.append(nombres[i]) else: if nombres[i]==nombres[i].lower(): listaB.append(nombres[i]) else: listaC.append(nombres[i].upper())</pre>			
A	<pre>listaA= ['lucas', 'juan', 'rafa'] listaB= ['Ana', 'Inés', 'Joaquín', 'Mariana'] listaC= ['JIMENA']</pre>		A
B	<pre>listaA= ['lucas', 'JIMENA', 'Joaquín', 'mariana'] listaB= ['Ana', 'juan', 'rafa'] listaC= ['INÉS']</pre>		B
C	<pre>listaA= ['inés', 'mariana'] listaB= ['RAFA', 'JUAN', 'ANA', 'LUCAS'] listaC= ['JIMENA', 'Joaquín']</pre>		C
D	<pre>listaA= ['lucas', 'Ana', 'juan', 'rafa'] listaB= ['inés', 'mariana'] listaC= ['JIMENA', 'JOAQUÍN']</pre>	X	D

Ejercicio 4 – Tema 1		1 Pto	
Hay un programa que no logra cargar ordenadamente en forma descendente la lista pares y ordenadamente en forma ascendente la lista resto con los elementos de numeros que corresponda En pares debe quedar [10, 2] y en resto [-6, -3.36, 0.25, 5] ¿Cuál falla? Notas: La serie de los pares son números naturales (enteros y mayores que 0) divisibles por 2 El método sort() ordena una lista; la opción reverse=True permite ordenarla de forma descendente lis=[6,-1,0] lis.sort() -> lis=[-1, 0, 6] lis.sort(reverse=True) -> [6, 0, -1] El método insert() permite agregar un elemento en una lista en cualquier posición lis=[1, 2, 3] lis.insert(1,8) -> lis=[1, 8, 2, 3] El método copy() genera una copia de la lista lis=[6,0] lis1=lis.copy() -> hay dos listas que en ese momento tienen los mismos datos El método remove() quita la primer ocurrencia de un elemento en una lista lis=[6,-1,0,5,6] lis.remove(6) -> lis=[-1, 0, 5, 6]			
A	<pre>numeros=[-3.36,5,10,2,0.25,-6] pares=numeros.copy() resto=[] for num in numeros: if num<=0: pares.remove(num) resto.append(num) elif num%2!=0: pares.remove(num) resto.append(num) resto.sort() pares.sort(reverse=True)</pre>		A
B	<pre>numeros=[-3.36,5,10,2,0.25,-6] pares=[] resto=[] for num in numeros: if num<=0 or num%2==0: resto.append(num) else: numeros.remove(num) pares=numeros</pre>	X	B
C	<pre>numeros=[-3.36,5,10,2,0.25,-6] pares=[] resto=[] for num in numeros: if num<=0: resto.append(num) elif num%2!=0: resto.append(num) else: pares.append(num) resto.sort() pares.sort(reverse=True)</pre>		C
D	<pre>numeros=[-3.36,5,10,2,0.25,-6] numeros.sort() pares=[] resto=[] for num in numeros: if num<=0: resto.append(num) elif num%2!=0: resto.append(num) else: pares.insert(0,num)</pre>		D

Ejercicio 5 – Tema 1		1 Pto	
¿Qué devuelve la función fun? <pre>def fun(lis): vocales='AEIOUÁÉÍÓÜ' cant=0 for pal in lis: pal=pal.upper() for voc in vocales: cant+=pal.count(voc) return cant txts=['FUNDA','sábana','Acolchado'] print(fun(txts))</pre>			
A	False		A
B	9	X	B
C	20		C
D	True		D

Ejercicio 6 – Tema 1		2 Ptos	
¿Qué muestra por pantalla el siguiente programa?			
<pre>cartel='literal' linea='' # vacía for i in range(0,len(cartel),3): linea+=cartel[i].upper()+' '*2 print(linea) linea=' ' # 1 blanco for i in range(1,len(cartel),3): linea+=cartel[i].upper()+cartel[i+1].upper()+' ' print(linea) linea='' # vacía for i in range(0,len(cartel),3): linea+=cartel[i].upper()+' '*2 print(linea)</pre>			
Notas: El operador * aplicado a una string es repetición Ej: 'hola'*3 -> 'holaholahola'			
A	L E L I T R A L E L		A
B	LI ER L T A L TE AL		B
C	L E L IT RA L E L	X	C
D	L I T E R A L L L L I T E R A L		D

Ejercicio 7 – Tema 1			2 Ptos
¿Cuál programa muestra el siguiente texto como salida? Juan, durante enero, sólo comió peras Notas: El método capitalize() devuelve la string en minúsculas, salvo la primera que se pone en mayúsculas Ej: 'vamos a las DOCE. Qué lindo!'.capitalize() -> 'Vamos a las doce. qué lindo!'			
A	<pre>cartel1='durante' cartel2='peras, manzanas, naranjas y otras frutas...' nombre='JUAN' mes=4 meses=('','enero','febrero','marzo','abril') salida=cartel1.capitalize()+' '+meses[mes].upper() salida+=', '+nombre+' sólo comió '+cartel2[17:] print(salida)</pre>		A
B	<pre>cartel1='durante' cartel2='peras, manzanas, naranjas y otras frutas...' nombre='JUAN' mes=0 meses=('','enero','febrero','marzo','abril') salida=cartel1.capitalize()+' '+meses[mes].upper() salida+=nombre+' sólo comió '+cartel2 print(salida)</pre>		B
C	<pre>cartel1='durante' cartel2='peras, manzanas, naranjas y otras frutas...' nombre='juan' mes=4 meses=('','enero','febrero','marzo','abril') salida=nombre+', '+cartel1+' '+meses[mes].lower() salida+=' sólo comió '+cartel2[17] print(salida.upper())</pre>		C
D	<pre>cartel1='durante' cartel2='peras, manzanas, naranjas y otras frutas...' nombre='juan' mes=1 meses=('','enero','febrero','marzo','abril') salida=nombre+', '+cartel1+' '+meses[mes]+' ', salida+='sólo comió '+cartel2[:5] print(salida.capitalize())</pre>	X	D

Ejercicio 8 – Tema 1		2 Ptos	
¿Cuál es la versión incorrecta (la única que no funciona adecuadamente) de la función edita ? El programa debe mostrar:			
taNTo va el cáNTaro a la fueNTe, que al final se roMPe			
<pre>def edita (...) : - - - frase='TANTO VA el cántaro a la Fuente, que al final SE Rompe' grupos=('MB','MP','NT','TR') print(edita(frase,grupos))</pre>			
Notas: El método replace() permite cambiar todas las apariciones de una string por otra dentro de un texto Ej: 'llueve a las doce. Qué llueva, qué llueva!'.replace('llue','yu') -> "yuve a las doce. Qué yuva, qué yuva!"			
A	<pre>def edita(t,grupos): grMin=[] for gr in grupos: grMin.append(gr.lower()) t=t.lower() for gr in grMin: t=t.replace(gr,gr.upper()) return t</pre>		A
B	<pre>def edita(t,grupos): t=t.lower() for gr in grupos: t=t.replace(gr.lower(),gr) return t</pre>		B
C	<pre>def edita(grupos,t): t=t.upper() for gr in grupos: t=t.replace(gr,gr.lower()) return t.lower()</pre>	X	C
D	<pre>def edita(t,grupos): t=t.upper() for i in range(len(grupos)): t=t.replace(grupos[i],str(i)) t=t.lower() for i in range(len(grupos)): t=t.replace(str(i),grupos[i]) return t</pre>		D

Ejercicio 9 – Tema 1		2 Ptos	
¿Cuál condición es equivalente a la siguiente? Para verificar el valor de verdad de cada una puedes usar los siguientes datos: valores=(11,22,33,55) num1=44 num2=300 (num1%11==0 and num1 in valores) or (num2%11!=0 and num2>0 and num2<100)			
A	num2 in range(1,100) or num2%11>0 or num1 in valores or num1%11<1		A
B	(num1%11==0 and num1 not in valores) or (num2>0 and num2<100)		B
C	(num2 in range(1,100) and num2%11>0) or (num1 in valores and num1%11<1)	X	C
D	(num2 in range(100,1000) and num2%11!=0) or (num1 not in valores and num1%11!=0)		D