

Ejercicio 0108		
Seleccione la opción correcta		
1.	Usar un lenguaje de programación no nos permite usar lenguaje coloquial para comunicarnos con la computadora.	
2.	Usar un lenguaje de programación es imprescindible para crear algoritmos de calidad.	
3.	Usar un lenguaje de programación nos asegura que nuestros programas funcionarán correctamente.	
4.	Usar un lenguaje de programación requiere que cumplamos con ciertas reglas.	
5.	Usar un lenguaje de programación nos ayuda a que nuestros programas tengan más líneas.	
6.	Ninguna es correcta.	
7.	1, 2 y 3 son correctas.	
8.	2 y 5 son correctas.	
9.	1 y 4 son correctas.	X

Ejercicio 0207		
¿Para qué sirven los parámetros en las funciones?		
1.	Los parámetros nos indican qué hace la función y determinan cuántos valores se devolverán	
2.	Los parámetros determinan la cantidad de líneas de una función, cuántos más parámetros, más líneas	
3.	Sirven para darle información a la función y que pueda resolver un problema	X
4.	Parámetros de una función es un concepto que no existe	
5.	1 y 2 son correctas	
6.	2 y 3 son correctas	

Ejercicio 0308		
¿Cuáles de los siguientes son programas válidos si se quiere mostrar por pantalla el valor de v y w?		
1.	<pre>print("V vale", v) print("W vale", v)</pre>	
2.	<pre>print("V vale", 4) print("W vale", 5)</pre>	
3.	<pre>def algo(v): print("V vale " + str(v)) def algo(w): print("W vale " + str(w)) v = 6 w = 8 algo(v) algo(w)</pre>	
4.	<pre>def algo_1(v): print("V vale " + str(v)) def algo_2(w): print("W vale " + str(w)) v = 6 w = 8 algo_1(v) algo_2(w)</pre>	
5.	<pre>def algo(letra, x): print(letra + " vale " + str(x)) v = 6 w = 8 algo("V", v) algo("W", w)</pre>	
6.	Ninguno es válido	
7.	1 y 2 son correctas	
8.	4 y 5 son correctas	X

Ejercicio 0406		
Las condiciones en una estructura selectiva se componen (entre otras cosas) de...		

1.	Funciones	
2.	Operadores de asignación	
3.	Operadores aritméticos	
4.	Operadores relacionales (o de comparación)	
5.	2, 3 y 4 son correctas	
6.	1, 2, 3 son correctas	
7.	1, 3 y 4 son correctas	X
8.	Ninguna es correcta	

Ejercicio 0504		
¿Cuál de los siguientes bloques de código no es correcto?		
1.	<pre>if txt[len(txt)] not in 'sl': print('El texto termina con un caracter que no es s ni l') if txt[len(txt)] in 'sl': print('El texto termina con s o l')</pre>	X
2.	<pre>if txt[len(txt)-1]=='s' or txt[len(txt)-1]=='l': print('El texto termina con s o l') elif txt[len(txt)-1]!='s' or txt[len(txt)-1]!='l': print('El texto termina con un caracter que no es s ni l')</pre>	
3.	<pre>if txt[len(txt)-1] in 'sl': print('El texto termina con s o l') else: print('El texto termina con un caracter que no es s ni l')</pre>	
4.	<pre>if txt[0] not in ('s', 'l'): print('El texto inicia con un caracter que no es s ni l') if txt[0] in 'sl': print('El texto inicia con s o l')</pre>	

Ejercicio 0603		
¿Cuál de las siguientes es una estructura iterativa?		
1.	if	
2.	def	
3.	for	X
4.	Ninguna opción es correcta	

Ejercicio 0707		
¿Cuántas veces imprime 'hola' el siguiente bloque de código? <pre>i=0 while i<7: print('hola') i = i + 1</pre>		
1.	Imprime 6 veces	
2.	Imprime 7 veces	X
3.	Nunca va a parar de imprimir	
4.	Ese bloque de código está mal sintácticamente, va a dar un error antes de ejecutarlo	

Ejercicio 0805		
Si tengo la siguiente lista: <pre>lista = [2,4,6,8]</pre> Y quiero eliminar el elemento "8" del final y quedarme con una lista que sea: <pre>lista = [2,4,6]</pre> ¿Cuáles de los siguientes bloques de código elimina correctamente el último elemento?		
1.	<pre>lista[3] = None</pre>	
2.	<pre>lista.pop()</pre>	X
3.	<pre>lista = lista - [8]</pre>	
4.	<pre>lista = lista - 1</pre>	

Ejercicio 0906		
----------------	--	--

Se quiere hacer una función que devuelva true si la lista recibida contiene al menos un número menor o igual a 5. ¿Cuál de las siguientes funciones hace lo pedido?		
1.	<pre>def contiene_menor_o_igual(lista): tiene_menor_o_igual = False for elemento in lista: if elemento > 5: tiene_menor_o_igual = True return tiene_menor_o_igual</pre>	
2.	<pre>def contiene_menor_o_igual(lista): tiene_menor_o_igual = False for elemento in lista: if elemento <= 5: tiene_menor_o_igual = True return tiene_menor_o_igual</pre>	X
3.	<pre>def contiene_menor_o_igual(lista): tiene_menor_o_igual = True for elemento in lista: if elemento <= 5: tiene_menor_o_igual = False return tiene_menor_o_igual</pre>	
4.	<pre>def contiene_menor_o_igual(lista): tiene_menor_o_igual = True for elemento in lista: if elemento > 5: tiene_menor_o_igual = False return tiene_menor_o_igual</pre>	

Ejercicio 1003		
¿Cuál de los siguientes problemas podría ser resuelto usando la función map?		
1.	Quiero eliminar todos los elementos que tengan más de cinco letras en una lista de strings	
2.	Quiero conseguir el promedio de una lista de números	
3.	Quiero agregar un elemento nuevo a una lista	
4.	Quiero restarle uno a todos los elementos de una lista de números	X

Ejercicio 1101	
Dado el siguiente fragmento de código	
<pre>def funcion_misteriosa(palabra): if(len(palabra)>5): return palabra.upper() return palabra lista = ['Hola','cómo','están','todos?'] print(list(map(funcion_misteriosa ,lista)))</pre>	
¿Qué se imprimirá por pantalla cuando se ejecute?	
1.	['Hola', 'cómo', 'están', 'todos?']
2.	['Hola', 'cómo', 'están', 'TODOS?']
3.	['Hola', 'cómo', 'ESTÁN', 'TODOS?']
4.	['Hola', 'CÓMO', 'ESTÁN', 'TODOS?']

Ejercicio 1206		
¿Cuál de los siguientes problemas podría ser resuelto usando la función filter?		
1.	Quiero agregar un elemento más al final de una lista de apellidos	
2.	Quiero obtener el mínimo elemento de una lista de edades	
3.	Quiero multiplicar por dos todos los elementos de una lista de temperaturas pero solo si son menores a 20	
4.	Quiero obtener los elementos que comienzan con 'A' de una lista de nombres	X

Ejercicio 1305	
Dado el siguiente fragmento de código	

<pre>def transformar(persona): return (persona[1] % 2 == 0) lista=[('Logan', 75), ('Kendall', 40), ('Roman', 35), ('Shiv', 30), ('Tom', 34), ('Connor', 54)] lista_transformada=list(map(transformar,lista)) print(lista_transformada)</pre>		
¿Qué se imprimirá por pantalla cuando se ejecute?		
1.	[False, True, False, True, True, True]	X
2.	[False, False]	
3.	[True, True, True, True]	
4.	[('Logan', 75), ('Roman', 35)]	
5.	[('Kendall', 40), ('Shiv', 30), ('Tom', 34), ('Connor', 54)]	
6.	[]	