

Ejercicio 0106		
Seleccione la opción correcta		
1.	Usar un lenguaje de programación 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 y si cumplimos con ellas, podemos usar un lenguaje coloquial.	
5.	Usar un lenguaje de programación nos ayuda a que nuestros programas tengan más líneas.	
6.	Ninguna es correcta.	X
7.	1 y 3 son correctas.	
8.	2, 4 y 5 son correctas.	

Ejercicio 0206		
¿Cuál es la identificación de la siguiente función? def incrementar_valores(numero_1, numero_2): numero_1 = numero_1 + 5 numero_2 = numero_2 + 10 return (numero_1, numero_2)		
1.	return (numero_1, numero_2)	
2.	def incrementar_valores(numero_1, numero_2)	
3.	def nombre(int, int)	
4.	La identificación de una función es un concepto que no existe	X
5.	1 y 2 son correctas	

Ejercicio 0307		
¿Cuáles de los siguientes son programas válidos si se quiere mostrar por pantalla el valor de v y w?		
1.	v = 1 w = 2 algo(v, w) print("V vale", v) print("W vale", v)	
2.	def algo(v, w): print("V vale " + str(v)) print("W vale " + str(w)) v = 6 w = 8 algo(v)	
3.	def algo(v, w): print("V vale "+ str(v)) print("W vale " + str(w)) v = 6 w = 8 algo(v, v)	
4.	def aglo(v, w): print("V vale " + str(v)) print("W vale " + str(w)) algo(2, 4)	
5.	def algo(): print("V vale " + str(v)) print("W vale " + str(w)) v = 6 w = 6 algo(v, w)	
6.	Ninguno es válido	X

Ejercicio 0407		
¿Para qué sirven las expresiones?		

1.	Se pueden usar operadores de asignación adentro de la condición de un if	X
2.	La estructura if sirve para encapsular un bloque de código, al cual puedo pasarle parámetros y obtener una respuesta	X
3.	Se pueden usar operadores aritméticos adentro de la condición de un if	X
4.	La estructura if sirve para iterar, repitiendo un bloque de código múltiples veces hasta cumplir con una condición	X
5.	Ninguna es correcta	X

Hubo un error en la redacción de la consigna, se dará cómo correcta cualquiera de las opciones.

Ejercicio 0503		
¿Cuál de los siguientes bloques de código no es correcto?		
1.	<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>	
2.	<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
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[len(txt)-1] not in ('s', 'l'): print('El texto termina con un caracter que no es s ni l') if txt[len(txt)-1] in 'sl': print('El texto termina con s o l')</pre>	

Ejercicio 0607		
¿Para qué tipo de estructura se usa un for?		
1.	Iterativa	X
2.	Selectiva	
3.	Argumentativa	
4.	Es una función	

Ejercicio 0703		
¿Cuántas veces imprime 'hola' el siguiente bloque de código? <pre>i=0 while i<=7: print('hola')</pre>		
1.	Imprime 7 veces	
2.	Imprime 8 veces	
3.	Nunca va a parar de imprimir	X
4.	El bloque de código está mal sintácticamente, va a dar un error antes de ejecutarlo	

Ejercicio 0803		
¿Qué va a imprimir el siguiente bloque de código al ejecutarse? <pre>lista = [2,4,6] print(lista[-1])</pre>		
1.	2	
2.	4	
3.	6	X
4.	El programa no funciona	

Ejercicio 0904		
Se quiere hacer una función que devuelva true si la lista recibida contiene todos elementos impares. ¿Cuál de las siguientes funciones hace lo pedido?		
1.	<pre>def todos_impares(lista): tiene_todos_impares = False for elemento in lista:</pre>	

	<pre>if elemento % 2 == 0: tiene_todos_impares = True return tiene_todos_impares</pre>	
2.	<pre>def todos_impares(lista): tiene_todos_impares = False for elemento in lista: if elemento % 2 != 0: tiene_todos_impares = True return tiene_todos_impares</pre>	
3.	<pre>def todos_impares(lista): tiene_todos_impares = True for elemento in lista: if elemento % 2 != 0: tiene_todos_impares = False return tiene_todos_impares</pre>	
4.	<pre>def todos_impares(lista): tiene_todos_impares = True for elemento in lista: if elemento % 2 == 0: tiene_todos_impares = False return tiene_todos_impares</pre>	X

Ejercicio 1001		
¿Cuál de los siguientes problemas podría ser resuelto usando la función map?		
1.	Quiero eliminar todos los elementos pares de una lista de números	
2.	Quiero conseguir la suma de todos los elementos de una lista de números	
3.	Quiero borrar el último elemento de una lista	
4.	Quiero sumarle uno a todos los elementos de una lista	X

Ejercicio 1108		
<p>Dado el siguiente fragmento de código</p> <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> <p>¿Qué se imprimirá por pantalla cuando se ejecute?</p>		
1.	['Hola!', 'CÓMO', 'están', 'todos?']	
2.	['Hola!', 'CÓMO', 'están', 'TODOS?']	X
3.	['HOLA!', 'CÓMO', 'están', 'TODOS?']	
4.	['Hola!', 'Cómo', 'están', 'TODOS?']	

Ejercicio 1201		
¿Cuál de los siguientes problemas podría ser resuelto usando la función filter?		
1.	Quiero emprolijar los nombres de una lista, que queden en minúsculas iniciando con mayúsculas (title())	
2.	Quiero quedarme sólo con los elementos que comienzan con 'A' de una lista de apellidos	X
3.	Quiero borrar solo el primer elemento de una lista de nombres	
4.	Quiero obtener el máximo elemento de una lista de edades	

Ejercicio 1306		
<p>Dado el siguiente fragmento de código</p> <pre>def transformar(persona): return (persona[1], (persona[1] > 56)) lista=[('Logan', 75), ('Kendall', 40), ('Roman', 35), ('Shiv', 30), ('Tom', 34), ('Connor', 54)]</pre>		

lista_filtrada=list(map(transformar,lista)) print(lista_filtrada)		
¿Qué se imprimirá por pantalla cuando se ejecute?		
1.	[(True, 'Logan'), (False, 'Kendall'), (False, 'Roman'), (False, 'Shiv'), (False, 'Tom'), (False, 'Connor')]	
2.	[(False, True), (True, False), (True, False), (True, False), (True, False), (True, False)]	
3.	[('Logan', True), ('Kendall', False), ('Roman', False), ('Shiv', False), ('Tom', False), ('Connor', False)]	
4.	[(True, True), (False, False), (False, False), (False, False), (False, False), (False, False)]	
5.	[(75, True), (40, False), (35, False), (30, False), (34, False), (54, False)]	X
6.	[]	