

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	DOCENTE (nombre y apellido):
TEL:	
AULA:	

Duración del examen: 1 hora 20 minutos. Completar los datos personales con letra clara, mayúscula e imprenta. El examen consta de 13 preguntas de opción múltiple. Cada pregunta tiene una y solo una respuesta correcta. Se debe contestar marcando con una X la opción elegida.

Ejercicio 0102			
Se tiene el siguiente código ticket = [{"nombre": "Pan", "cantidad": 5, "precio": 100}, {"nombre": "Vino", "cantidad": 3, "precio": 700}, {"nombre": "Chocolate", "cantidad": 2, "precio": 300}] imprimir_total_por_producto(ticket) Queremos imprimir, para cada elemento de la lista, el nombre y el valor total de las cantidades compradas. Elegir la implementación de la función que cumpla con lo pedido			
1.	def imprimir_total_por_producto(): for p in prods: print(p["nombre"], p["cantidad"] * p["precio"])		1.
2.	def imprimir_total_por_producto(prods): print(p["nombre"], p["cantidad"] * p["precio"])		2.
3.	def imprimir_total_por_producto(prods): for nombre, cant, precio in prods: print(nombre, cant * precio)		3.
4.	def imprimir_total_por_producto(prods): for p in prods: print(p[0], p[1] * p[2])		4.
5.	def imprimir_total_por_producto(prods): for p in prods: print(p["nombre"], p["cantidad"] * p["precio"])	X	5.

Ejercicio 0202			
¿Cómo sabemos qué hace una función definida dentro de un programa?			
1.	Por el orden de los argumentos		1.
2.	Por el tipo de dato que devuelve		2.
3.	Analizando el cuerpo de la función	X	3.
4.	Por los nombres de los argumentos		4.
5.	Por el resultado que devuelve		5.

Ejercicio 0302			
¿Cuál sería una versión correcta del programa para indicar si un número ingresado es primo?			
1.	<pre>def primo(n): resul=True for i in range(2,n//2+1): if n%i==0: resul=False num=int(input('Núm: ')) if primo(num): resultado='es primo' else: resultado='no es primo' print(num,resultado)</pre>		1.
2.	<pre>def primo(n): resul=True for i in range(2,n//2+1): if n%i==0: resul=False return resul num=int(input('Núm: ')) resul=primo(num) if resul==False: resultado='es primo' else: resultado='no es primo' print(num,resultado)</pre>		2.
3.	<pre>def primo(n): resul=True for i in range(2,n//2+1): if n%i==0: resul=False return i num=int(input('Núm: ')) resul=primo(num) if resul: resultado='es primo' else: resultado='no es primo' print(num,resultado)</pre>		3.
4.	<pre>def primo(n): resul=True for i in range(2,n//2+1): if n%i==0: resul=False return resul num=int(input('Núm: ')) if primo(num): resultado='es primo' else: resultado='no es primo' print(num,resultado)</pre>	X	4.

Ejercicio 0402			
Cual de las siguientes operaciones no se puede realizar dentro de un if?			
1.	Llamar a una función		1.
2.	Hacer un return		2.
3.	Hacer otro if		3.
4.	Hacer una asignación		4.
5.	Ninguna de las opciones anteriores es correcta. Todas las opciones se pueden hacer dentro de un if	X	5.

Ejercicio 0502			
¿Cuál es el código correcto para saber si un par (día, mes) es válido o no? (No se consideran los años bisiestos)			
1.	<pre>d=int(input('Día: ')) m=int(input('Mes: ')) if ((m in (1,3,5,7,8,10,12) or d in range(1,32)) or(m in(4,6,9,11) or d in range(1,31)) or(m==2 or d in range(1,29))): print(d,'/',m,' es una fecha correcta',sep='') else: print(d,'/',m,' no es una fecha correcta',sep='')</pre>		1.
2.	<pre>d=int(input('Día: ')) m=int(input('Mes: ')) if m in range(1,13) and d in range(1,32): print(d,'/',m,' es una fecha correcta',sep='') else: print(d,'/',m,' no es una fecha correcta',sep='')</pre>		2.
3.	<pre>d=int(input('Día: ')) m=int(input('Mes: ')) if ((m in (1,3,5,7,8,10,12) and d in range(1,32)) or(m in(4,6,9,11) and d in range(1,31)) or(m==2 and d in range(1,29))): print(d,'/',m,' es una fecha correcta',sep='') else: print(d,'/',m,' no es una fecha correcta',sep='')</pre>	X	3.
4.	<pre>d=int(input('Día: ')) m=int(input('Mes: ')) if (m in (1,3,4,5,6,7,8,9,10,11,12) and d in range(1,32) and m==2 and d in range(1,29)): print(d,'/',m,' es una fecha correcta',sep='') else: print(d,'/',m,' no es una fecha correcta',sep='')</pre>		4.

Ejercicio 0602			
Un Bucle while			
1.	Sólo se puede usar cuando se necesita un número fijo de repeticiones		1.
2.	Se utiliza únicamente para iterar sobre una secuencia de elementos		2.
3.	Puede usarse para lo mismo que un bucle for	X	3.
4.	Siempre se ejecuta al menos una vez		4.

Ejercicio 0702			
¿Qué muestra el siguiente código?			
<pre>final='' txt='Una cosa más, para terminar' for i in range(5,len(txt),2): final+=txt[i] print(final)</pre>			
1.	Ua oa a, aa eia		1.
2.	oams aatria	X	2.
3.	osa más, para terminar		3.
4.	258111417202326		4.

Ejercicio 0802			
¿Cuál de estos códigos <u>no</u> muestra la siguiente secuencia: 3 5 6 45 9 ?			
1.	<pre>lista=[88,-4,0,1,3,5,6,45,9] print(lista[4:])</pre>		1.
2.	<pre>lista=[88,-4,0,1,3,5,6,45,9] for i in range(len(lista)): if i>3: print(lista[i],end=' ')</pre>		2.

3.	lista=[88,-4,0,1,3,5,6,45,9] for i in range(1,6): print(lista[-i],end=' ')	X	3.
4.	lista=[88,-4,0,1,3,5,6,45,9] for i in range(4,len(lista)): print(lista[i],end=' ')		4.

Ejercicio 0902			
¿Cuál de las siguientes funciones devuelve True si la lista recibida tiene por lo menos dos palabras que comienzan con p?			
1.	def funcion(lis): listaFiltrada = [] for pal in lis: if pal[0].lower()=='p': listaFiltrada.append(pal) return listaFiltrada		1.
2.	def funcion(lis): listaFiltrada = [] for pal in lis: if pal[0].lower()=='p': listaFiltrada.append(pal) if len(listaFiltrada)>1: return True else: return False	X	2.
3.	def funcion(lis): cont=0 for pal in lis: if pal[0].lower()=='p': cont+=1 if cont==len(lis): return True else: return False		3.
4.	def funcion(lis): if lis[0].lower()=='p': return True else: return False		4.

Ejercicio 1002			
Con respecto a listas y diccionarios. ¿Cuál de las siguientes afirmaciones es falsa?			
1.	Ambos nos permiten guardar múltiples elementos		1.
2.	Los elementos que almacenan pueden ser de distinto tipo de dato		2.
3.	En ambos puedo usar un for para recorrer sus elementos		3.
4.	Ambos tipos son una secuencia	X	4.

Ejercicio 1102			
¿Qué se imprime con el siguiente código? def funcion(p): if not p[0].lower() in 'abcdefg': return p return p[::-1].upper() lista=['la', 'araña','que','pica'] nueva= list(map(funcion,lista)) string = '' for el in nueva: string += f"{el} " print(string)			

1.	la araña que pica		1.
2.	acip euq añara al		2.
3.	acip ueq AÑARA al		3.
4.	la AÑARA que pica	X	4.

Ejercicio 1202			
¿Con qué valor va a terminar la variable 'nueva'?			
<pre>def funcion(n): if n==str(n): return True return False lista=[25,3.88,-0.66,-99,0] nueva=list(filter(funcion,lista))</pre>			
1.	[]	X	1.
2.	[-0.66, 0]		2.
3.	[-0.66,-99]		3.
4.	[3.88,-0.66]		4.

Ejercicio 1302			
¿Qué devuelve esta función map?			
<pre>def funcion(n): n=str(n) return (n*5)[:5] lista=[123,345,0,987] relleno=list(map(funcion,lista))</pre>			
1.	[123,345,0,987]		1.
2.	[]		2.
3.	['12312','34534',' 00000','98798']	X	3.
4.	[12312,34534, 00000 ,98798]		4.