

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 0101			
Se tiene el siguiente código			
<pre>estudiantes = [{"nombre": "Camila", "apellido": "Torres", "nota": 10}, {"nombre": "Martin", "apellido": "Flores", "nota": 3}, {"nombre": "Ana", "apellido": "Suárez", "nota": 7}] imprimir_apellidos_y_notas(estudiantes)</pre>			
Queremos imprimir, para cada elemento de la lista, el apellido y la nota. Elegir la implementación de la función que cumpla con lo pedido			
1.	<pre>def imprimir_apellidos_y_notas(lista_est): for e in lista_est: print(e[1], e[2])</pre>		1.
2.	<pre>def imprimir_apellidos_y_notas(lista_est): for e in lista_est: print(e["apellido"], e["nota"])</pre>	X	2.
3.	<pre>def imprimir_apellidos_y_notas(lista_est): print(e["apellido"], e["nota"])</pre>		3.
4.	<pre>def imprimir_apellidos_y_notas(lista_est): for e in lista_est: print(e["nombre"], e["nota"])</pre>		4.
5.	<pre>def imprimir_apellidos_y_notas(): for e in lista_est: print(e["apellido"], e["nota"])</pre>		5.

Ejercicio 0201			
¿Cómo sabemos qué hace una función definida dentro de un programa?			
1.	Por la cantidad de líneas que tiene la función		1.
2.	Toda función inicia obligatoriamente con un comentario aclarativo		2.
3.	Por los nombres de los argumentos		3.
4.	Entendiendo lo que hace el cuerpo de la misma	X	4.
5.	Por el resultado que devuelve		5.

Ejercicio 0301			
¿Cuál sería una versión correcta de 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 return resul num=int(input('Núm: ')) resul=primo(num) if resul==False: 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: ')) 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 0401			
Un Condicional o estructura if se puede usar			
0			
1.	Cuando necesito repetir bloques de código		1.
2.	Cuando necesito seguir caminos de acción diferentes para distintas situaciones	X	2.
3.	Cuando quiero reutilizar código		3.
4.	Para mostrar un dato por pantalla		4.
5.	Para evaluar una operación aritmética compleja con más de un operador y distintas precedencias		5.

Ejercicio 0501			
¿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) 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	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) 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>		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 0601			
Un Bucle while			
1.	Se puede emplear en cualquier situación que requiera un ciclo o estructura iterativa	X	1.
2.	Sólo deja de ejecutar el bloque de código dentro cuando se encuentra con un break		2.
3.	Sólo se puede usar cuando se necesita un número fijo de repeticiones		3.
4.	No permite emplear una condición compleja		4.

Ejercicio 0701			
¿Qué muestra el siguiente código? final='' txt='Una cosa más, para terminar' for i in range(2,len(txt),3): final+=txt[i] print(final)			
1.	a cosa más, para terminar		1.
2.	Una cosa más, para terminar		2.
3.	ao spaeir	X	3.
4.	126		4.

Ejercicio 0801			
¿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.	<pre>lista=[88,-4,0,1,3,5,6,45,9] for i in range(1,6): print(lista[-i],end=' ')</pre>	X	3.
4.	<pre>lista=[88,-4,0,1,3,5,6,45,9] for i in range(4,len(lista)): print(lista[i],end=' ')</pre>		4.

Ejercicio 0901			
¿Cuál de las siguientes funciones devuelve True si la lista recibida tiene por lo menos dos palabras que comienzan con p?			
1.	<pre>def funcion(lis): resp=False for pal in lis: if pal[0].lower()=='p': resp=True return resp</pre>		
2.	<pre>def funcion(lis): cont=0 for pal in lis: if pal[0].lower()=='p': cont+=1 if cont>1: return True else: return False</pre>	X	
3.	<pre>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</pre>		
4.	<pre>def funcion(lis): if lis[0].lower()=='p': return True else: return False</pre>		

Ejercicio 1001			
¿Cuándo conviene usar un diccionario?			
1.	Cuando se deben guardar pocos datos		1.
2.	Cuando se quiere mantener la información ordenada		2.
3.	Cuando se necesita localizar datos sin saber su posición, por un dato o clave asociado	X	3.
4.	Cuando se necesita invertir el orden varias veces durante el proceso		4.

Ejercicio 1101			
¿Qué se imprime con el siguiente código?			
<pre>def funcion(p): if p[0].lower() in 'aeiouáéíóú': return p return p[::-1] lista=['ÁLVARO','uma','Pía','sara'] nueva= list(map(funcion,lista)) string = '' for el in nueva: string += f"{el} "</pre> <pre>print(string)</pre>			

1.	álvaro uma pía sara		1.
2.	ÁLVARO uma PÍA sara		2.
3.	ÁLVARO uma aíP aras	X	3.
4.	Oravlá amu aíp aras		4.

Ejercicio 1201			
¿Con qué valor va a terminar la variable 'nueva'?			
<pre>def funcion(n): if n==int(n): return True return False lista=[25,3.88,-0.66,-99,0] nueva=list(filter(funcion,lista))</pre>			
1.	[25,-99,0]	X	1.
2.	[]		2.
3.	[-0.66,-99]		3.
4.	[3.88,-0.66]		4.

Ejercicio 1301			
¿Qué devuelve esta función map?			
<pre>def funcion(n): n=str(n) return (n*10)[:10] lista=[25,388,0,1266] relleno=list(map(funcion,lista))</pre>			
1.	[2500,3880,0,12660]		1.
2.	[]		2.
3.	['2525252525','388388388388',' 0000000000','126612661266']		3.
4.	['2525252525','3883883883',' 0000000000','1266126612']	X	4.