

PENSAMIENTO COMPUTACIONAL (90)

**UBAXXI
TEMA 7**

EXAMEN: PRIMER PARCIAL

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

Duración del examen: 1:30h.

- ✓ Escribir claramente el nombre en todas las páginas.
- ✓ El examen consta de 10 preguntas de opción múltiple.
- ✓ Cada pregunta tiene una y sólo una respuesta correcta.
- ✓ Las respuestas seleccionadas deben consignarse en la siguiente matriz de opciones.
- ✓ **Sólo se considerarán las respuestas anotadas en la matriz.**
- ✓ Las preguntas de la 1 a la 7 inclusive permiten acumular 1 punto (si son correctas), de la 8 a la 10 cada una acumula 2 puntos o 0.
- ✓ La nota final se calcula de acuerdo a la siguiente función:

Puntos	1 o 2	3 o 4	5 o 6	7	8	9	10	11	12	13
Nota	1	2	3	4	5	6	7	8	9	10

Matriz de Respuestas

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 1 Pto	Ej 7 1 Pto	Ej 8 2 Ptos	Ej 9 2 Ptos	Ej 10 2 Ptos	
1											1
2											2
3											3
4											4

¡ATENCIÓN! Las respuestas sólo se considerarán válidas si se encuentran en la matriz. De haber diferencias entre la opción seleccionada en el ejercicio y en la matriz, se considerará como válida esta última.

0107 - 1 Pto		
Si curso=1 y ape='Paz' , Cuál programa evalúa más condiciones?:		
1	<pre> if curso<3: if ape in ('Clarins','Lavarten'): print('Universitario Ciclo Inicial') elif curso==2: if ape in ('Durand','Sandex'): print('Universitario Ciclo Inicial') elif curso==1: if ape in ('Daneri','Paz'): print('Ingresante') elif ape =='Paz': print('Universitario avanzado') </pre>	X 1
2	<pre> if curso<3: if ape in ('Paz','Clarins','Lavarten'): print('Universitario Ciclo Inicial') elif curso==2: if ape in ('Durand','Sandex'): print('Universitario Ciclo Inicial') elif curso==1: if ape in ('Daneri','Paz'): print('Ingresante') elif ape =='Paz': print('Universitario avanzado') </pre>	2
3	<pre> if (ape=='Paz' or ape=='Pérez')and curso in range(1,4): print('Universitario Ciclo Inicial') elif ape =='Paz': print('Universitario avanzado') elif curso>3: print('Universitario Avanzado') else: print('No está anotado') </pre>	3
4	<pre> ape='Paz' if ape=='Pérez')and curso in range(1,4): print('Universitario Ciclo Inicial') if ape=='Paz')and curso in range(1,4): print('Universitario Ciclo Inicial') if ape=='Antelo')and curso in range(1,4): print('Universitario Ciclo Inicial') </pre>	4

0207 - 1 Pto		
¿Cuál de las siguientes condiciones no es equivalente al resto? Sugerencia: Probá para los siguientes valores... a=180 , b=3 , base=30		
1	(base*b>a or a%2==0) and base%2==0	1
2	(a<b*base and base%2==0) or (a%2==0 and base%2==0)	2
3	a%2<1 and a<b*base and base%2<1	X 3
4	(a%2<1 or a<b*base) and base%2<1	4

0307 - 1 Pto			
<p>¿Cuál de los siguientes programas no deja la lista c de la siguiente manera? c=['luis', 'luis', 'luis', 'luis']</p> <p>Notas: El método extend() le agrega al final una lista a otra Ej: a=[1,0] a.extend([2,3]) -> [1,0,2,3]</p> <p>El operador + concatena listas, respetando el orden Ej: a=[1,0,2,1] b=[5,5] a+b -> [1,0,2,1,5,5] y b+a -> [5,5,1,0,2,1]</p> <p>El operador * repite listas Ej: a=[1,0,2] a*2 -> [1,0,2,1,0,2]</p>			
1	<pre>a=['luis'] b=['pedro', 'ana', 'juan'] c=[] for nom in b: c.extend(a) for nom in a: c.extend(a)</pre>		1
2	<pre>a=['luis'] b=['pedro', 'ana', 'juan'] d=len(a)+len(b) c=[] for nom in a: c=c+b</pre>	X	2
3	<pre>a=['luis'] b=['pedro', 'ana', 'juan'] d=len(a)+len(b) c=a*d</pre>		3
4	<pre>a=['luis'] b=['pedro', 'ana', 'juan'] c=a*len(b+a)</pre>		4

0407 - 1 Pto			
¿Cuál de los siguientes códigos muestran por pantalla ningún \$?			
1	<pre>for i in range(9,5, -3): print('\$') i=8 while i<9: print('\$') i+=1</pre>		1
2	<pre>for i in range(9,5, -1): j=i while j<i: print('\$') j+=2 i=3 while i<9: print('\$') i+=1</pre>		2
3	<pre>for letra in 'hola qué tal?': if letra in '*=\$': print('\$')</pre>	X	3
4	<pre>for i in range(1,9,3): for j in range(3):</pre>		4

	<code>print('\$')</code>		
	0507 - 1 Pto		
	<p>¿Cuál será la salida por pantalla del siguiente programa?</p> <pre> porcentajes={'%b':'10%', '%m':'30%', '%a':'70%'} meses={'&1':'Enero', '&2':'Febrero', '&3':'Marzo', '&4':'Abril'} temporada={'\$1':'Verano 2024', '\$2':'Invierno 2023', '\$3':'Verano 2023'} txt='Ofertas \$2, de &2 a &4\n%a off en zapatillas\nOfertas \$1, sólo &3\n%b off en remeras' txtEditado=txt for cgo in porcentajes: if cgo in txtEditado: txtEditado=txtEditado.replace(cgo,porcentajes[cgo]) for cgo in meses: if cgo in txtEditado: txtEditado=txtEditado.replace(cgo,meses[cgo]) for cgo in temporada: if cgo in txtEditado: txtEditado=txtEditado.replace(cgo,temporada[cgo]) print(txtEditado) </pre> <p>Nota: El carácter de control '<code>\n</code>' produce una bajada de línea en la pantalla</p> <p>Ej: <code>a='Yo\nsoy\nasí'</code> <code>print(a) -> Yo</code> <code>soy</code> <code>así</code></p>		
1	Invierno 2023FebreroAbril 70% Verano 2024Marzo 10%		1
2	Ofertas Invierno 2023, de Febrero a Abril 70% off en zapatillas Ofertas Verano 2024, sólo Marzo 10% off en remeras	X	2
3	OFERTAS \$2, DE &2 A &4 %A OFF EN ZAPATILLAS OFERTAS \$1, SÓLO &3 %B OFF EN REMERAS		3
4	OFERTAS , DE A OFF EN ZAPATILLAS OFERTAS , SÓLO OFF EN REMERAS		4

	0607 - 1 Pto		
	<p>¿Qué devuelve fun() si recibe la siguiente lista: <code>['cero','dos','cuatro','seis','ocho']</code> y el número 3 ?</p> <pre> def fun(lista,num): if num in range(0,9,2): resultado=lista[num//2].upper() else: resultado='Indefinido' return resultado </pre>		
1	<code>''</code>		1
2	<code>['cero']</code>		2

3	'Indefinido'	X	3
4	'dos'		4

0707 - 1 Pto			
<p>¿Cuál de los códigos no es equivalente al siguiente?</p> <pre>i=0 numeros=[1, -10, 7, 3] mayor=max(numeros) while i<len(numeros): print(mayor-numeros[i]) i+=1</pre> <p>Nota: La función max() devuelve el mayor valor de una lista</p> <p>Ej: a=[1,0,5,2] max(a) -> 5</p> <p>La función abs() devuelve el valor absoluto de un número</p> <p>Ej: abs(-10) -> 10 y abs(10) -> 10</p> <p>El método reverse() invierte una lista</p> <p>Ej: a=[1,0] a.reverse() -> a=[0,1]</p> <p>Un índice negativo señala una posición del fondo o final hacia adelante en una lista</p> <p>Ej: a=[1,0,3,4,2] a[-1] -> 2 y a[-4] -> 0</p>			
1	<pre>numeros=[1, -10, 7, 3] mayor=max(numeros) for num in numeros: print(abs(mayor-num))</pre>		1
2	<pre>numeros=[1, -10, 7, 3] mayor=numeros[0] i=len(numeros) while i>0: i-=1 print(mayor-numeros[i])</pre>	X	2
3	<pre>numeros=[1, -10, 7, 3] numeros.reverse() mayor=max(numeros) for i in range(1, len(numeros)+1): print(mayor-numeros[-i])</pre>		3
4	<pre>numeros=[1, -10, 7, 3] mayor=numeros[0] for num in numeros[1:]: if num>mayor: mayor=num for i in range(len(numeros)): print(mayor-numeros[i])</pre>		4

0807 - 2 Ptos			
<p>¿Qué función masLarga hay que usar para que el siguiente programa identifique la palabra con más letras de lista?</p> <pre>def masLarga(...): - - - #PPal lista=['perro', 'vaca', 'caballo', 'tero', 'perdiz'] posicion=masLarga(lista) print('La palabra más larga de:', lista, 'es', lista[posicion])</pre> <p>El programa debería mostrar por pantalla lo siguiente:</p> <p>La palabra más larga de: ['perro', 'vaca', 'caballo', 'tero', 'perdiz'] es caballo</p> <p>Nota: La función max() devuelve el valor máximo de una lista</p> <p>Ej: max([1,2,3]) -> 3 y max(['ana','elena']) -> 'elena'</p>			
1	<pre>def masLarga(): mayor='' for i in range(len(l)): if l[i]>mayor: mayor=l[i] return mayor</pre>		1
2	<pre>def masLarga(l): mayor=0 for i in range(len(l)): if len(l[i])>mayor: mayor=len(l[i]) posi=i return posi</pre>	X	2
3	<pre>def masLarga(): largos=[] for pal in l: largos.append(len(pal)) mayor=max(largos) posi=largos.index(mayor) return largos, posi</pre>		3
4	<pre>def masLarga(l, mayor): mayor=max(l) posi=l.index(mayor)</pre>		4

0907 - 2 Ptos			
<p>¿Qué muestra por pantalla el siguiente programa?</p> <pre>x=10 y=5.5 z=1 k=0.5 calculo= (x-y)/k*(z+k) print(calculo)</pre>			
1	9.0		1
2	13.5	X	2
3	0.0		3
4	1.0		4

1007 - 2 Ptos			
Indique cuál es la salida correcta del siguiente programa:			
<pre>def conRepe(pal): i=0 marca=0 while i<len(pal) and marca==0: if pal.count(pal[i])>1: marca=1 else: i+=1 return marca==1 ultimos=pal[1:] return not ultimos.islower() or not pal[0].isupper() verduras=['PAPA', 'BATATA', 'LECHUGA', 'KALE', 'CHAUCHA'] i=0 while i<len(verduras): if conRepe(verduras[i]): verduras.pop(i) else: i+=1 print(verduras)</pre>			
Nota: El método <i>pop()</i> elimina el elemento de la lista que está en la posición indicada en el argumento			
Ej: <i>a=[1,2,3,4,5]</i> <i>a.pop(2) -> a=[1,2,4,5]</i>			
1	['LECHUGA', 'KALE']	X	1
2	['papa', 'batata', 'chaucha']		2
3	[]		3
4	['PAPA', 'kale']		4