

PENSAMIENTO COMPUTACIONAL (90)

UBA XXI
TEMA 8

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:

Punt os	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.

	0108 - 1 Pto		
	Si curso=1 y ape='Paz' , Cuál programa evalúa menos 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>		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>	X	3
4	<pre>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

	0208 - 1 Pto		
	¿Cuál de las siguientes condiciones no es equivalente al resto? Sugerencia: Probá para los siguientes valores... a=180 , b=3 , base=80		
1	a%2==1 and a<b*base and base%2<1	X	1
2	(base*b>a or a%2==1) and base%2==0		2
3	(a<b*base and base%2!=1) or (a%2!=0 and base%2==0)		3
4	(a%2>0 or a<b*base) and base%2==0		4

	0308 - 1 Pto		
	<p>¿Cuál de los siguientes programas no deja la lista c de la siguiente manera? c= ['pedro', 'ana', 'luis', 'pedro', 'ana', '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=['elena'] b=['pedro','ana','luis'] largo=len(b)-len(a) c=b*largo</pre>		1
2	<pre>a=['elena'] b=['pedro','ana','luis'] c=a*len(b)</pre>	X	2
3	<pre>a=['elena'] b=['pedro','ana','luis'] c=[] for i in range(1,(len(b)-len(a))+1): c.extend(b)</pre>		3
4	<pre>a=['elena'] b=['pedro','ana','luis'] c=[] largo=len(b)-len(a) for i in range(largo): c=c+b</pre>		4

	0408 - 1 Pto		
	¿Cuál de los siguientes códigos muestran por pantalla exactamente 4 \$?		
1	<pre>for i in range(9,5,-1): j=i while j<i: print('\$') j+=2 i=7 while i<9: print('\$') i+=1</pre>		1
2	<pre>for letra in 'Me llamo Ana': if letra=='P': print('\$') for i in range(2): for j in range(1,3): print('\$')</pre>	X	2
3	<pre>for letra in 'hola qué tal?': if letra in '*=\$': print('\$')</pre>		3
4	<pre>for letra in 'Me llamo Ana': if letra not in 'aA': print('\$')</pre>		4

0508 - 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='Descuentos \$3, de &3 a &4\n%m off en jeans\nDescuentos \$1, de &2 a &3\n%a off en medias' 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 'n' produce una bajada de línea en la pantalla</p> <p>Ej: a='Yo\nsoy\nasí' print(a) -> Yo soy así</p>			
1	DESCUENTOS ---, DE --- A --- --- OFF EN JEANS DESCUENTOS ---, DE --- A --- --- OFF EN MEDIAS		1
2	30%70%FebreroMarzoAbrilVerano 2024Verano 2023		2
3	Descuentos Verano 2023, de Marzo a Abril 30% off en jeans Descuentos Verano 2024, de Febrero a Marzo 70% off en medias	X	3
4	DESCUENTOS \$3, DE &3 A &4 %M OFF EN JEANS DESCUENTOS \$1, DE &2 A &3 %A OFF EN MEDIAS		4

0608 - 1 Pto			
<p>¿Qué devuelve fun() si recibe la siguiente lista: ['cero','dos','cuatro','seis','ocho'] y el número 4 ?</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	'Indefinido'		1
2	'CUATRO'	X	2
3	' '		3

4	'ocho'		4
---	--------	--	---

	0708 - 1 Pto		
	<p>¿Cuál de los códigos no es equivalente al siguiente?</p> <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> <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=numeros[3] i=0 while i>0: i-=1 print(mayor-numeros[i])</pre>	X	1
2	<pre>numeros=[1, -10, 7, 3] numeros.reverse() mayor=max(numeros) for i in range(1, len(numeros)+1): print(mayor-numeros[-i])</pre>		2
3	<pre>numeros=[1, -10, 7, 3] mayor=max(numeros) for num in numeros: print(abs(mayor-num))</pre>		3
4	<pre>i=0 numeros=[1, -10, 7, 3] mayor=max(numeros) while i<len(numeros): print(mayor-numeros[i]) i+=1</pre>		4

	0808 - 2 Ptos		
	<p>¿Qué función masCorta hay que usar para que el siguiente programa identifique la palabra con menos letras de lista?</p> <pre>def masCorta(...): - - - #PPa1 lista=['perro', 'vaca', 'caballo', 'pez', 'perdiz'] posicion=masCorta(lista) print('La palabra más corta de:', lista, 'es', lista[posicion])</pre> <p>El programa debería mostrar por pantalla lo siguiente:</p> <p>La palabra más corta de: ['perro', 'vaca', 'caballo', 'pez', 'perdiz'] es pez</p> <p>Nota: La función min() devuelve el valor mínimo de una lista</p> <p>Ej: min([1,2,3]) -> 1 y min(['ana','elena']) -> 'ana'</p>		
1	<pre>def masCorta(lista,menor): menor=len(l[0]) posi=0 for i in range(1,len(l)): if len(l[i])<menor: menor=len(l[i]) posi=i</pre>		1
2	<pre>def masCorta(lista): menor=l[0] posi=len(l) for i in range(len(l)): if l[i]<menor: menor=l[i] posi=i return l</pre>		2
3	<pre>def masCorta(l): largos=[0]*len(l) for i in range(len(l)): largos[i]=len(l[i]) menor=min(largos) return largos.index(menor)</pre>	X	3
4	<pre>def masCorta(l, largos,menor): largos=[0]*len(l) for i in range(len(l)): largos[i]=len(l[i]) menor=min(largos) return l.index(menor)</pre>		4

	0908 - 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	-6.25		1
2	0.0		2

3	13.5	X	3
4	1.0		4

1008 - 2 Ptos

Indique cuál es la salida correcta del siguiente programa:

```
def sinRepe(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==0

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 sinRepe(verduras[i]):
        verduras.pop(i)
    else:
        i+=1
print(verduras)
```

Nota:

El método **pop()** elimina el elemento de la lista que está en la posición indicada en el argumento

Ej:

a=[1,2,3,4,5]

a.pop(2) -> a=[1,2,4,5]

1	['kale', 'lechuga']		1
2	['PAPA', 'BATATA', 'CHAUCHA']	X	2
3	[]		3
4	['Chaucha', 'Kale', 'Lechuga', 'Batata', 'Papa']		4