

PENSAMIENTO COMPUTACIONAL (90)

.UBA XXI

TEMA 4

EXAMEN: Julio 2024

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:

nota(puntos) = (puntos//7+1)%2*(puntos+1)//2 +(puntos//7%2)* (puntos-3)

Función nota(puntos) tabulada:

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.

0104 – 1 Pto			
¿Cuál de los siguientes códigos evalúa más condiciones si material='algodón' , largo=0.5 y tipo='fina' ?			
1	<pre>if (tipo=='fina' and material=='plástico') or largo<1: print('Soporta peso moderado') elif largo<2 or material=='plástico': if largo>=0.5 and tipo=='fina': print('Soporta peso medio') elif material in('X76','Hilo') and tipo=='fina': print('Baja resistencia')</pre>		1
2	<pre>if tipo=='media' or material=='plástico': print('Soporta peso moderado') elif largo<2 or material=='plástico': if largo>=0.5 and tipo=='fina': print('Soporta peso medio') elif material in('X76','Hilo') and tipo=='fina': print('Baja resistencia')</pre>		2
3	<pre>if tipo=='media' or material=='plástico': print('Soporta peso moderado') elif largo>2 and material=='plástico': print('Soporta peso medio') elif largo<1.5 or material=='acero': print('Soporta peso alto') elif material=='algodón' and largo>5: print('Soporta peso bajo') elif largo>=0.5 and largo<3 and material=='algodón': print('Soporta peso alto') else: print('Baja resistencia')</pre>	X	3
4	<pre>if largo>=0.5 and material in ('U25','algodón'): print('Soporta peso alto') elif tipo=='media' or material=='plástico': print('Soporta peso moderado') elif largo>2 and material=='plástico': print('Soporta peso medio') elif largo<1.5 or material=='acero': print('Soporta peso alto') else: print('Baja resistencia')</pre>		4

0204 – 1 Pto			
¿Cuál programa no muestra exactamente 4 *?			
1	<pre>for letra in 'Mafalda': if letra in 'AaFf': print('*') personaje='Susanita' while personaje in ('Felipe','Manolito'): print('*') personaje='Felipe'</pre>		1
2	<pre>for elem in (2,4,1,12,3,4,6,2,10,8): if elem%2==0: for i in range(1,2): print('*')</pre>	X	2
3	<pre>i=1 while i<8: j=3 while j>2: print('*') j-=1 i+=2</pre>		3
4	<pre>for i in range(19,17,-1): print('*') for j in range(13,3): print('*') for i in range(1): for k in range(2,4): print('*')</pre>		4

0304 – 1 Pto

Dado el siguiente DataFrame *stock*:

	Código	Descripción	Existencia	Pr Unit	Unidad
0	25	Silicona transparente	5.0	2300	pomo
1	3	Clavos autoperforantes	230.0	600	gramo
2	14	Lija nro 6	25.0	235	unidad
3	11	Lija nro 3	36.0	235	unidad
4	5	Tornillos Philips ½	NaN	980	unidad
5	107	Tornillo común 3/4	320.0	670	gramo
6	29	Martillo carpintero	NaN	8700	unidad
7	102	Cal viva	1400.0	270	gramo

¿Qué sentencia produce el siguiente resultado?

Unidad	
unidad	4
gramo	3
pomo	1

1	stock[stock['Existencia'].isnull()]		1
2	stock[stock['Unidad']=='gramo']		2
3	stock.head(2)		3
4	stock['Unidad'].value_counts()	X	4

0404 – 1 Pto

¿Cuál es la función **suprime** adecuada para el siguiente programa?

```
def suprime(...):  
    -  
    -  
    -  
  
#Ppal  
pal1='osadía'  
pal2='OsA'  
pal3='DÍA'  
print(suprime(pal2,pal1))  
print(suprime(pal3,pal1))
```

La salida debería ser:
día
osa

Nota:
El método **index()** devuelve la posición donde inicia un texto dentro de otro
Ej:
'Una complejidad Absoluta'.index('pleji') -> 7

El método **pop()** recibe una posición y elimina ese elemento en una lista
Ej:
[5,0,1,2].pop(2) -> [5,0,2]

El método **join()** devuelve un texto con los elementos (str) de una lista
Ej:
' '.join(['a','e','o']) -> 'a-e-o'

El método **replace()** devuelve un texto con el primer argumento remplazado por el segundo
Ej:
'Living la vida loca'.replace('vi','VU')-> 'LiVUng la VUda loca'

El método **count()** devuelve la cantidad de apariciones del argumento en una secuencia
Ej:
'Living la vida loca'.count('vi')-> 2

1	<pre>def suprime(txt1,txt2): resp=txt2 if txt2 in txt1: txt1.replace(txt2, '') return resp</pre>		1
2	<pre>def suprime(): if txt2.count(txt1)==1: txt2=txt2.replace(txt1, '') return txt2</pre>		2
3	<pre>def suprime(txt2): resp=txt2 if txt1.lower() in txt2.lower(): lista=list(txt2) inicia=txt2.lower().index(txt1.lower()) for i in range(len(txt1)): lista.pop(inicia) resp=' '.join(lista)</pre>		3
4	<pre>def suprime(txt1,txt2): resp=txt2 if txt1.upper() in txt2.upper(): lista=list(txt2) inicia=txt2.upper().index(txt1.upper()) for i in range(len(txt1)): lista.pop(inicia) resp=' '.join(lista) return resp</pre>	X	4

	0504 – 1 Pto		
	<p>¿Qué muestra el siguiente programa?</p> <pre>a=['enero',' ','marzo',' ','mayo',' ','julio', 'agosto',' ','octubre',' ','diciembre'] b=['ENE','FEB','MAR','ABR','MAY','JUN', 'JUL','AGO','SEP','OCT','NOV','DIC'] cortos=['MAY','febrero','meyor'] for c in cortos: if c[:3].upper() in b: posi=b.index(c[:3].upper()) if a[posi]!=' ': print(a[posi]) else: print('No tiene 31 días') else: print('No es un mes')</pre> <p>Nota: El método index() devuelve la posición donde se encuentra un elemento dentro de una lista Ej: [0,0,1,0,2].index(1) -> 2</p> <p>Se puede seleccionar la primera parte de una string haciendo s[:x] que toma los primeros caracteres hasta antes del de la posición x Ej: 'tornavías'[:4] -> 'torn'</p>		
1	mayo No tiene 31 días No es un mes	X	1
2	MAY		2
3	may feb mey		3
4	ENERO AGOSTO DICIEMBRE		4

0604 – 1 Pto

En el siguiente programa:

```
def selecciona(...):  
    -  
    -  
    -  
  
#Ppal  
palabras=['astucia','AZULADO','perMEABLE','Disponible']  
adjetivos=list(filter(selecciona,palabras))  
print(adjetivos)
```

La función **selecciona()** debe indicar si la palabra recibida es un adjetivo (terminado en ada, ado, able o ible). De ese modo el programa mostrará el siguiente resultado:

```
['AZULADO', 'perMEABLE', 'Disponible']
```

¿Cuál versión de **selecciona()** **no** realiza lo esperado?

1	<pre>def selecciona(pal): pal=pal.lower() termina=('ada','ado','able','ible') adj=False for t in termina: if t in pal: adj=True return adj</pre>		1
2	<pre>def selecciona(pal): esta=0 termina=('ada','ado','able','ible') i=0 while i<len(termina) and esta==0: if termina[i] in pal.lower(): esta+=1 i+=1 return esta==1</pre>		2
3	<pre>def selecciona(pal): esta=False for t in ('ada','ado','able','ible'): if t in pal.lower(): esta=True return esta</pre>		3
4	<pre>def selecciona(pal): esta=False termina=('ada','ado','able','ible') i=0 while i<len(termina): if termina[i] in pal.lower(): esta=False i+=1 return esta</pre>	X	4

0704 – 1 Pto

¿Cuál de los siguientes programas muestra el contenido de una lista desde el principio hasta la primera aparición u ocurrencia de un número? Si el número no está, debe mostrar la lista completa.

Nota:

El método **index()** devuelve la posición de un elemento en una secuencia. Si el elemento **no está** produce **ValueError** y aborta la ejecución

Ej:

[2,2].index(3) -> ValueError

[2,2].index(2) -> 0

1	<pre>num=0 lista=[1,1,4,5,6,2] indice=lista.index(num) try: print(lista) except ValueError: print(lista[0:indice])</pre>		1
2	<pre>num=0 lista=[1,1,4,5,6,2] try: indice=lista.index(indice) print(lista[0:indice]) elif ValueError: print(lista[0])</pre>		2
3	<pre>num=0 lista=[1,1,4,5,6,2] try: indice=lista.index(num) print(lista[0:indice]) except ValueError: print(lista)</pre>	X	3
4	<pre>num=0 lista=[1,1,4,5,6,2] if: indice=num.index(lista) print(lista[0:indice]) else: print(lista)</pre>		4

0804 – 2 Ptos

¿Qué muestra el siguiente programa?

```
def convierte(t):  
    verbo=t[:len(t)-4]+'aDO'  
    return verbo.lower()  
  
adjetivos=['divisaBle','RAZONable','compensABLE','ALcanzable']  
participios=list(map(convierte,adjetivos))  
print(participios)
```

Nota:
Se puede seleccionar la primera parte de una string haciendo **s[:x]** que toma los primeros caracteres hasta antes del de la posición **x**

Ej:
'tornavías'[:4] -> 'torn'

1	['divisado', 'razonado', 'compensado', 'alcanzado']	X	1
2	DIVISRAZONCOMPENSALCANZ		2
3	['DivisadO', 'RazonadO', 'CompensadO', 'AlcanzadO']		3
4	(' ', ' ', ' ', ' ')		4

0904 – 2 Ptos			
<p>Dado el siguiente programa:</p> <pre>def abre (arch,modo) : return open (arch,modo) arch1=abre('carrera1.txt',...) #carrera1.txt listaAl=arch1.readlines() arch1.close() lisParc=[] carre=listaAl[0] for alu in listaAl[1:]: datos=alu.split(',') lisParc.append(datos[1]+' '+datos[0]+';'+carre) arch1=abre('carrera2.txt',...) #carrera2.txt listaAl=arch1.readlines() arch1.close() carre=listaAl[0] for alu in listaAl[1:]: datos=alu.split(',') lisParc.append(datos[1]+' '+datos[0]+';'+carre) arch2=abre('parcial.txt',...) #parcial.txt for alu in lisParc: arch2.write(alu) arch2.close()</pre> <p>Que arma un listado para un parcial con alumnos de 2 carreras</p> <p>Contenido de carrera1.txt: Química Analía,Ricardes,2021 Juliana,Ortesano,2023 Pedro,Fuentes,2023</p> <p>Contenido de carrera2.txt: Industrial Mariano,Quimsey,2024 Dante,Drexler,2022 Ignacio,Artez,2023 Raquel,Venturini,2024</p> <p>¿Cuáles deberían ser los modos de apertura para cada archivo de forma que el programa funcione correctamente?</p>			
1	carrera1.txt modo 'r' carrera2.txt modo 'r' parcial.txt modo 'w'	X	1
2	carrera1.txt modo 'r' carrera2.txt modo 'r+' parcial.txt modo 'r'		2
3	carrera1.txt modo 'a' carrera2.txt modo 'a' parcial.txt modo 'a'		3
4	carrera1.txt modo 'w+' carrera2.txt modo 'a' parcial.txt modo 'r'		4

1004 – 2 Ptos			
<p>¿Qué salida produce el siguiente programa?</p> <pre>verduras={1:'papa',2:'zanahoria',3:'lechuga', 4:'tomate',5:'cebolla',6:'arvejas'} ensaladas=['mixta','rusa','de la casa','candice'] ingredientes=[[0,3],[0,4],[0,5],[2,1], [2,4],[2,2],[1,1],[1,2],[1,6]] carta={} for par in ingredientes: ensa=ensaladas[par[0]] lleva=verduras[par[1]] if ensa.upper() in carta: carta[ensa.upper()].append(lleva) else: carta[ensa.upper()]=[lleva] print('Ensaladas') for ensa in carta: print(ensa,':') print(*carta[ensa])</pre> <p>Nota: Si se coloca * delante de una secuencia en un print, se muestran todos los elementos de la misma, uno al lado del otro y separados por espacio.</p> <p>Ej: print(*['una','dos','tres']) -> una dos tres</p>			
1	Ensaladas mixta de la casa rusa candice		1
2	Ensaladas candice: mixta : lechuga de la casa : papa rusa : papa		2
3	Ensaladas lechuga: mixta tomate: mixta de la casa cebolla: mixta papa: de la casa rusa zanahoria: de la casa rusa arvejas: rusa		3
4	Ensaladas MIXTA : lechuga tomate cebolla DE LA CASA : papa tomate zanahoria RUSA : papa zanahoria arvejas	X	4



Talón de Control para el Alumno

	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