

PENSAMIENTO COMPUTACIONAL (90)

UBA XXI

TEMA 1

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.

0101 – 1 Pto			
¿Cuál de los siguientes códigos evalúa menos condiciones si borrable=False , trazo='fino' y color='azul' ?			
1	<pre>if not borrable: if color in ('negro','azul','verde','rojo'): if trazo=='fino': print('Lapicera') else: print('Marcador') if borrable==False and trazo=='grueso': print('Fibrón') else: print('Lápiz')</pre>		1
2	<pre>if not borrable and trazo=='fino': if color in ('negro','azul','verde','rojo'): print('Lapicera') else: print('Marcador trazo fino') elif borrable==False and trazo=='grueso': print('Fibrón') else: print('Lápiz')</pre>	X	2
3	<pre>if not borrable and color not in ('naranja','violeta'): if trazo!='grueso': print('Lapicera') if borrable: print('Lápiz') if borrable==False and trazo=='grueso': print('Fibrón') if not borrable or color=='fucsia': print('Resaltador')</pre>		3
4	<pre>if borrable or color in ('naranja','amarillo','violeta'): if trazo=='grueso': print('Resaltador') if borrable: print('Lápiz') else: if trazo=='grueso': print('Lápiz Especial') elif color=='fucsia': print('Resaltador') elif trazo in ('fino','medio') or color=='azul': print('Lapicera')</pre>		4

0201 – 1 Pto			
¿Cuál programa muestra exactamente 4 *?			
1	<pre>for i in range(2,6,2): for j in range(5,5,-1): print('*') for i in range(0,3,3): print('*')</pre>		1
2	<pre>i=1 while i<=4: j=2 while i+j<4: print('*') j+=1 i+=1</pre>		2
3	<pre>for i in range(2,6): for j in range(5,4,-1): print('*') for i in range(6,3): print('*')</pre>	X	3
4	<pre>i=1 while i<=3: for letra in 'Mafalda': print('*') i+=1</pre>		4

0301 – 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	
gramo	1400.0
pomo	5.0
unidad	36.0

1	stock[['Pr Unit', 'Código']]		1
2	stock[stock['Pr Unit'].between(100, 400)]		2
3	stock[stock['Existencia'].isnull()]		3
4	stock.groupby('Unidad')['Existencia'].max()	X	4

0401 – 1 Pto

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

```
def calcula(...):  
    -  
    -  
    -  
  
#Ppal  
num1=(3,4,-5,10)  
num2=(1,1,-3)  
print(calcula('+',num1))  
print(calcula('*',num2))
```

La salida debería ser:

12

-3

1	<pre>def calcula(op,operacion): acum=str(operacion[0]) i=1 while i<len(operacion): acum=acum+op+str(operacion[i]) i+=1</pre>		1
2	<pre>def calcula(op,operacion): if op=='+': acum=0 for num in operacion: acum+=num else: acum=1 for num in operacion: acum*=num return acum</pre>	X	2
3	<pre>def calcula(op): if op=='+': acum=1 for num in operacion: acum+=num else: acum=0 for num in operacion: acum*=num return op</pre>		3
4	<pre>def calcula(operacion,op): acum=0 for num in operacion: acum+=num return op,acum</pre>		4

0501 – 1 Pto

¿Qué muestra el siguiente programa?

```
a=['jujuy','salta','tucumán','catamarca',  
  'neuquén','mendoza']  
b=['JJY','SLT','TCM','CTC','NQN','MDZ']  
cgos=['TDF','CTC']  
for cg in cgos:  
    if cg in b:  
        posi=b.index(cg)  
        print(a[posi].upper())  
    else:  
        print('No sé')
```

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

1	No sé NO SÉ		1
2	CTC TDF		2
3	Tucumán salta		3
4	No sé CATAMARCA	X	4

0601 – 1 Pto			
En el siguiente programa:			
<pre>def selecciona (...): - - - #Ppal palabras=['divisaBle', 'RAZONar', 'compensAado', 'ALcance'] adjetivos=list(filter(selecciona,palabras)) print(adjetivos)</pre>			
¿Cuál versión de selecciona() permite que el mismo seleccione los adjetivos de la lista palabras y muestre lo siguiente?			
['divisaBle', 'compensAado']			
Nota:			
Los números usados como condición toman los valores de verdad de acuerdo a la siguiente función: 0 -> False y cualquier número distinto de 0 -> True			
Ej:			
if 2: -> evalúa verdadero			
if not 0: -> evalúa verdadero			
if 4>1 and 0: -> evalúa falso			
1	<pre>def selecciona(pal): termina=('ada','ado','able','ible') for t in termina: if t.lower() in pal.upper(): esta=False else: esta=True return esta</pre>		1
2	<pre>def selecciona(pal): esta=False termina=('ada','ado','able','ible') if pal in termina: esta=True return esta</pre>		2
3	<pre>def selecciona(pal): termina=('ada','ado','able','ible') t=termina[0] if t in pal.lower(): esta=True return True</pre>		3
4	<pre>def selecciona(pal): pal=pal.lower() termina=('ada','ado','able','ible') encontrada=0 for t in termina: if t in pal: encontrada+=1 return encontrada</pre>	X	4

0701 – 1 Pto			
¿Cuál de los siguientes programas realiza un ingreso validado de la edad de un asociado a una org que debe ser mayor de edad y menor de 80 años? El programa debe insistir hasta conseguir un ingreso correcto y evitar un fallo o error fatal ante un ingreso indebido			
1	<pre>ok=True while ok: try: edad=int(input('Edad Asociado: ')) if edad in range(18,80): ok=False else: print('Mayor de edad y menor a 80 años') except ValueError: print('Debe ser un entero') print(edad)</pre>	X	1
2	<pre>while True: try: edad=int(input('Edad Asociado: ')) if edad not in range(18,80): ok=False else: print('Mayor de edad y menor a 80 años') except ValueError: print('Debe ser un entero') print(edad)</pre>		2
3	<pre>try: edad=int(input('Edad Asociado: ')) if edad not in range(18,80): print('Mayor de edad y menor a 80 años') except ValueError: print('Debe ser un entero') print(edad)</pre>		3
4	<pre>edad=int(input('Edad Asociado: ')) while edad>18: edad=int(input('Edad Asociado: ')) print(edad)</pre>		4

0801 – 2 Ptos			
¿Qué muestra el siguiente programa?			
<pre>def convierte(t): term='able' return (t[:len(t)-1]+term).upper() verbos=['lamento', 'RAZONO', 'SELecciona', 'almaCENA'] adjetivos=list(map(convierte,verbos)) print(adjetivos)</pre>			
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	lamentoable		1
2	('almacena', 'selecciona', 'razono', 'lamento')		2
3	'ableableableable'		3
4	['LAMENTABLE', 'RAZONABLE', 'SELECCIONABLE', 'ALMACENABLE']	X	4

0901 – 2 Ptos			
Dado el siguiente programa: <pre>def abre (arch,modo): return open (arch,modo) arch1=abre('comi1.txt',...) #comi1.txt listaAl=arch1.readlines() arch1.close() lisParc=[] for alu in listaAl: datos=alu.split(',') lisParc.append(datos[2]+' '+datos[1]+' '+datos[0]+'\\n') arch1=abre('comi2.txt',...) #comi2.txt listaAl=arch1.readlines() arch1.close() for alu in listaAl: datos=alu.split(',') lisParc.append(datos[2]+' '+datos[1]+' '+datos[0]+'\\n') arch2=abre('pensamiento.txt',...) #pensamiento.txt for alu in lisParc: arch2.write(alu) arch2.close()</pre> Que arma un listado de alumnos para un parcial con las comisiones 1 y 2 Contenido de comi1.txt: 9878955,Caldes,Joaquín,Sistemas 1721058,Ayllander,Luciana,Industrial 3658455,Estensoro,Julián,Industrial Contenido de comi2.txt: 2298955,Keer,Patricio,Química 1902108,Collaneri,Ivo,Industrial 2136845,Zatto,Manuel,Sistemas ¿Cuáles deberían ser los modos de apertura para cada archivo de forma que el programa funcione correctamente?			
1	comi1.txt modo 'r' comi2.txt modo 'a' pensamiento.txt modo 'r'		1
2	comi1.txt modo 'r' comi2.txt modo 'r+' pensamiento.txt modo 'w'	X	2
3	comi1.txt modo 'w' comi2.txt modo 'w+' pensamiento.txt modo 'a'		3
4	comi1.txt modo 'r+' comi2.txt modo 'w' pensamiento.txt modo 'r'		4

1001 – 2 Ptos

¿Qué salida produce el siguiente programa?

```
colores={1:'Blanca',2:'Roja',3:'Rosa',
          4:'Amarilla',5:'Lila'}
flores=['rosa','margarita','gerbera','clavelina']
puestoSel=[[0,2],[1,1],[0,3],[0,4],
            [2,4],[2,2],[3,1],[1,5]]
variedad={}
for par in puestoSel:
    flor=flores[par[0]]
    color=colores[par[1]]
    if flor in variedad:
        variedad[flor].append(color)
    else:
        variedad[flor]=[color]
print('En el Puesto puedes elegir:')
for flor in variedad:
    for color in variedad[flor]:
        print(flor.upper(),color)
```

1	En el Puesto puedes elegir: ROSA Roja ROSA Rosa ROSA Amarilla MARGARITA Blanca MARGARITA Lila GERBERA Amarilla GERBERA Roja CLAVELINA Blanca	X	1
2	En el Puesto puedes elegir: 0 ROJA 0 ROSA 0 AMARILLA 1 BLANCA 1 LILA 2 AMARILLA 2 ROJA 3 BLANCA		2
3	En el Puesto puedes elegir: ROSA rosa ROJA rosa gerbera AMARILLA rosa gerbera BLANCA margarita clavelina LILA margarita		3
4	En el Puesto puedes elegir: blanca, 1, 3 roja, 0, 2 rosa, 0 amarilla, 0, 2 lila, 1		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