

PENSAMIENTO COMPUTACIONAL (90)

UBA XXI

TEMA 3

EXAMEN: SEGUNDO 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

iATENCIÓN! 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.

0103 – 1 Pto			
0203 – 1 Pto			
¿Cuál de los siguientes códigos valida adecuadamente que se ingrese un nombre sin tilde?			
¿Cuál es la validación correcta del siguiente programa?			
def nomMes(mes): meses=[1:'enero',2:'febrero',3:'marzo',4:'abril',5:'mayo',6:'junio',7:'julio',8:'agosto',9:'septiembre',10:'octubre',11:'noviembre',12:'diciembre'] return meses[mes].upper() voc='AEIOU' marca=False #PPal i=0 while not marca and i<len(voc): nombres=list(map(nomMes,meses)) if voc[i] in n.upper(): print(nombres) marca=True 1 i+=1 Nota: return marca El método upper() devuelve el texto todo en mayúsculas Ej: nom=input('Nombre sin acentos: ') a=tilde(nom) or not nom.isalpha(): a.upper()>'HOLA' Nombre sin acentos: '			
1	def tilde(n): voc='ÁÉÍÓÚ' for letra in voc: for letra in n.lower(): if letra in voc: marca=True else: marca=False return marca	X	1
2	for letra in voc: for letra in n.lower(): if letra in voc: marca=True else: marca=False return marca		2
3	#PPal nom=input('Nombre sin acentos: ') while tilde(nom) or not nom.isalpha(): nom=input('Nombre sin acentos: ')		3
3	def tilde(n): voc='ÁÉÍÓÚ' marca=False for letra in n.lower(): if letra in voc: marca=True return True #PPal nom=input('Nombre sin acentos: ') while tilde(nom) or not nom.isalpha(): nom=input('Nombre sin acentos: ')		3
4	def tilde(n): voc='ÁÉÍÓÚ' marca=False if n.lower() in voc: marca=True return marca #PPal nom=input('Nombre sin acentos: ') if nom.isalpha() or tilde(nom): print('Válido') else: print('Inválido')		4

0303 – 1 Pto		
¿Cuál es la salida del siguiente programa?		
<pre>def noLaborable(d): finde=(6,7) return d in finde #PPal # 1 es lunes y 7 es domingo dias=[1,5,4,6,2,7] sinBanco=list(filter(noLaborable,dias)) print(sinBanco)</pre>		
1	[1, 2]	1
2	[6, 7]	X 2
3	5	3
4	[]	4

0403 – 1 Pto ¿Qué contenido tendrá el archivo destacados.txt al finalizar la ejecución del programa si el archivo deportistas.txt tiene el siguiente contenido? Contenido de deportistas.txt: lionel,messi,futbol gaby,sabattini,tenis rafa,nadal,tenis chapu,nocioni,basquet Programa a ejecutarse: arch=open('deportistas.txt','r') lista=arch.readlines() arch.close() nombres=[] for dep in lista: depor=dep.split(',') nombres.append(depor[1]+'*****\n') arch=open('destacados.txt','w') arch.writelines(nombres) arch.close() Nota: El método split() devuelve una lista con las partes de un texto tomando como separador el argumento Ej: 'yo soy argentina'.split(' ') -> ['yo', 'soy', 'argentina']			
1	messi***** sabattini***** nadal***** nocioni*****	X	1
2	Rafa, Nadal, Tenis Chapu, Nocioni, Basquet		2
3	Lionel,, gaby,, rafa,, chapu,,		3
4	lionel,messi,futbol gaby,sabattini,tenis rafa,nadal,tenis chapu,nocioni,basquet		4

	0503 – 1 Pto		
¿Qué debería ser estructura para que la siguiente instrucción se ejecute sin problemas?			
dato=estructura['Hola']			
1	Un diccionario	X	1
2	Una lista de lista		2
3	Una tupla		3
4	Una string		4

0603 – 1 Pto

Para averiguar el camarote de un viaje en barco asignado a un pasajero se dispone del siguiente programa. ¿Cuál función impide que aborte la ejecución del mismo si se ingresa el nombre **Alicia**?

def cabina(...):
-
-
-

barco={'Ana':15,'Ulises':2,'Florencia':1,
 'Ignacio':15, 'Agustín':17,'Paulina':22}
print('Averiguá en qué camarote viaja')
pasajero=input('Nombre: ')
print(pasajero,cabina(barco,pasajero))

1	<pre>def cabina(nom): except: camarote=b[nom] cartel='viaja en camarote '+str(camarote) try: cartel='NO VIAJA EN ESTE BARCO' return cartel</pre>		1
2	<pre>def cabina(nom,b): camarote=b[nom] cartel='viaja en camarote '+str(camarote) return cartel</pre>		2
3	<pre>def cabina(b,nom): camarote=b[nom] try: cartel='viaja en camarote '+str(camarote) except: cartel='NO VIAJA EN ESTE BARCO' return nom</pre>		3
4	<pre>def cabina(b,nom): try: camarote=b[nom] cartel='viaja en camarote '+str(camarote) except: cartel='NO VIAJA EN ESTE BARCO' return cartel</pre>	X	4

0703 – 1 Pto							
Para el DataFrame vue de pandas, que contiene:							
	nombre	dni	fila	asiento	tarifa		
0	Paz, Orlando	25889145	14	C	flex		
1	Manzur, Jimena	15226223	11	D	promo		
2	Hermenegildo, Juan	19002136	6	A	promo		
3	Decquer, Mariqana	41100107	22	C	base		
4	Tasz, Martín	36214875	14	F	flex		
¿Qué contendrá vue1 después de la siguiente operación?							
vue1=vue[vue['asiento']!='C']							
1	nombre	dni	fila	asiento	tarifa	X	1
	1 Manzur, Jimena	15226223	11	D	promo		
	2 Hermenegildo, Juan	19002136	6	A	promo		
	4 Tasz, Martín	36214875	14	F	flex		
2	dni	tarifa					2
	0 25889145	flex					
	1 15226223	promo					
	2 19002136	promo					
	3 41100107	base					
	4 36214875	flex					
3	nombre	dni	tarifa				3
	0 Paz, Orlando	25889145	flex				
	1 Manzur, Jimena	15226223	promo				
	2 Hermenegildo, Juan	19002136	promo				
	3 Decquer, Mariqana	41100107	base				
4	nombre	dni	fila	asiento	tarifa		4
	0 Paz, Orlando	25889145	14	C	flex		

0803 – 2 Ptos			
¿Cuáles modos de apertura deben emplearse con el archivo <i>notas.txt</i> en el siguiente programa para que no salte error?			
<pre>arch=open('notas.txt',...) #primer open() filas=['1\n', '2\n', '3\n'] arch.writelines(filas) arch.close() arch=open('notas.txt',...) #segundo open() arch.write('Fin\n') arch.close()</pre>			
1	Primer open() 'r' Segundo open() 'r+'		1
2	Primer open() 'w' Segundo open() 'a'	X	2
3	Primer open() 'r' Segundo open() 'a'		3
4	Primer open() 'a' Segundo open() 'r'		4

0903 – 2 Ptos						
Para el DataFrame vue de pandas, que contiene:						
	nombre	dni	fila	asiento	tarifa	
0	Paz, Orlando	25889145	14	C	flex	
1	Manzur, Jimena	15226223	11	D	promo	
2	Hermenegildo, Juan	19002136	6	A	promo	
3	Decquer, Mariqana	41100107	22	C	base	
4	Tasz, Martín	36214875	14	F	flex	
¿Qué operación produce el siguiente resultado?						
6						
1	vue['fila'].min()					X 1
2	vue.loc[:, ['fila', 'asiento']]					2
3	vue[vue['tarifa'].isnull()]					3
4	vue.iloc[2:]					4

1003 – 2 Ptos

En el siguiente programa:

```
txt='Me picaron MAS mosquitos QUE EN EL VERANO'
traductor={'e':')','o': '%','a': '*','i': '/',
            'u': '@','m': '&', ' ': '+-' }
for letra in txt:
    if letra in traductor:
        ..... # línea a completar
print(txt)
```

Que traduce los caracteres del texto que están definidos en **traductor**. ¿Cuál debería ser la línea faltante en el código?

La salida final debería ser:

M) +-p/c*r%n+-MAS+-&%sq@/t%s+-QUE+-EN+-EL+-VERANO

1	txt.replace(traductor[letra], '1')		1
2	txt=txt.replace(letra,traductor[letra])	X	2
3	If letra in txt:		3
4	letra=traductor[letra]		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