

PENSAMIENTO COMPUTACIONAL (90)

UBA XXI

TEMA 7

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.

0107 – 1 Pto			
<p>¿Cuál de los siguientes códigos valida adecuadamente que ingrese un nombre sin <b>a</b> o <b>u</b>? Debe detectar el error y garantizar un dato válido <b>Ejs válidos: Inés, Pedro, JORGE</b> <b>Ejs inválidos: Ángelo, laura, 9emilio</b></p>			
1	<pre>def hayAU(n):     voc='auáúü'     marca=False     for v in voc:         if v in n.lower():             marca=True     return marca  #PPal nom=input('Nombre sin a ni u: ') while hayAU(nom) or not nom.isalpha():     nom=input('Nombre sin a ni u: ')</pre>	X	1
2	<pre>def hayAU(n):     voc='auáúü'     for v in voc:         if v in n:             marca=True         else:             marca=False     return marca  #PPal nom=input('Nombre sin a ni u: ') while hayAU(nom) and not nom.isalpha():     nom=input('Nombre sin a ni u: ')</pre>		2
3	<pre>voc='auáúü' nom=input('Nombre sin a ni u: ') for letra in nom:     if letra in voc:         print('Error de ingreso')</pre>		3
4	<pre>def hayAU(n):     voc='auáúüAUÀÜÜ'     marca=False     for v in voc:         if v in n:             marca=True         else:             marca=False  #PPal nom=input('Nombre sin a ni u: ') while hayAU(nom) or nom.isalpha():     nom=input('Nombre sin a ni u: ')</pre>		4

0207 – 1 Pto			
<p>¿Cuál es la salida correcta del siguiente programa?</p> <pre>def nomDia(d):     dias={1:'domingo',2:'lunes',3:'martes',4:'miércoles',           5:'jueves',6:'viernes',7:'sábado'}     return dias[d]  #PPal dias=[1,5,2,6,7] nombres=list(map(nomDia,dias)) print(nombres)</pre>			
1	['domingo']		1
2	['DOM', 'JUE', 'LUN', 'VIE', 'SÁB']		2
3	[1, 5, 2, 6, 7]		3
4	['domingo', 'jueves', 'lunes', 'viernes', 'sábado']	X	4

0307 – 1 Pto

11/6/2024

¿Cuál es la salida del siguiente programa?

Tema 7 – Pag 3

```
def mes31(m):  
    de30dias=(4,6,9,11)  
    return m not in de30dias and m!=2  
  
#PPal  
meses=[1,5,6,2,7]  
de31=list(filter(mes31,meses))  
print(de31)
```

1	[1, 5, 6, 2, 7]		1
2	'noviembre'		2
3	[1, 5, 7]	X	3
4	[]		4

			0407 – 1 Pto
	<p>¿Qué contenido tendrá el archivo <b>invitados.txt</b> al finalizar la ejecución del programa si el archivo <b>contactos.txt</b> tiene el siguiente contenido?</p> <p>Contenido de <b>contactos.txt</b>:</p> <p>Juan,álvarez,juanjo12@gmail.com,Colegio Ana Emilia,Paz,anitapaz@yahoo.com.ar,trabajo Rodo,García,rgarcia04@gmail.com,trabajo Elías,Fernández,elifer@gmail.com,barrio Walter,Demarco,wdemarco@mitrabajo.com.ar,colegio Ema,Derby,derbita34@gmail.com,TRABAJO</p> <p>Programa a ejecutarse:</p> <pre>arch=open('contactos.txt','r+') lista=arch.readlines() arch.close() invitados=[] for amigo in lista:     amigo=amigo.strip('\n')     a=amigo.split(',')     if a[3].lower()=='trabajo':         invitados.append(a[0]+' '+a[2]+'\\n') arch=open('invitados.txt','w') arch.writelines(invitados) arch.close()</pre> <p><b>Nota:</b> El método <b>split()</b> devuelve una lista con las partes de un texto tomando como separador el argumento <b>Ej:</b> <b>'yo soy argentina'.split(' ') -&gt; ['yo', 'soy', 'argentina']</b></p> <p>El método <b>strip()</b> devuelve el texto quitándole el argumento de los extremos <b>Ejs:</b> <b>'yo soy argentina!!'.strip('!! ') -&gt; 'yo soy argentina'</b> <b>'-nada-'.strip('-') -&gt; 'nada'</b></p>		
1	Juan,Álvarez,juanjo12@gmail.com Ana Emilia,Paz,anitapaz@yahoo.com.ar Rodo,García,rgarcia04@gmail.com Elías,Fernández,elifer@gmail.com Walter,Demarco,wdemarco@mitrabajo.com.ar Ema,Derby,derbita34@gmail.com		1
2	Juan,Álvarez,,COLEGIO Elías,Fernández,,BARRIO Walter,Demarco,,COLEGIO		2
3	Colegio Trabajo Trabajo Barrio Colegio Trabajo		3
4	Ana Emilia,anitapaz@yahoo.com.ar Rodo,rgarcia04@gmail.com Ema,derbita34@gmail.com	X	4

	0507 – 1 Pto		
¿Qué podría ser <b>estructura</b> para que la siguiente instrucción se ejecute sin problemas?  estructura[0]='primero'			
1	Una lista vacía		1
2	Una tupla con dos elementos		2
3	Un diccionario	X	3
4	La cadena nula ('')		4

0607 – 1 Pto

¿Cuál función devuelve el titular de una patente impidiendo que aborte la ejecución del programa si se ingresa la patente **jdd 003** ?

```
def quien(...):  
    -  
    -  
    -  
  
titulares={'AG 234 FF': 'Andreoli, Julieta',  
           'AC 217 DR': 'Vargas, Ulises',  
           'KDF 876': 'Fiquet, Paulo',  
           'LBD 976': 'Carassa, Emilia'}  
pat=input('Patente: ')  
print('El titular de', pat, quien(titulares, pat))
```

1	<pre>def quien(titulares, pat):     try:         cartel='es '+titulares[pat.upper()]     except:         cartel=' es desconocido'     return cartel</pre>	X	1
2	<pre>def quien():     except:         cartel='es '+titulares[pat.upper()]     else:         cartel=' es desconocido'     return cartel</pre>		2
3	<pre>def quien(titulares, pat):     cartel='es '+titulares[pat.upper()]     return cartel</pre>		3
4	<pre>def quien(titulares, pat):     nombre=titulares[pat.upper()]     try:         cartel='es '+nombre     return cartel</pre>		4

0707 – 1 Pto							
Para el DataFrame <b>vue</b> de pandas, que contiene:							
	fila	asiento	zona	pax			
0	5	A	4	tripulación			
1	16	A	2	None			
2	16	B	2	Marcelo Uriondo			
3	16	C	2	Delsy Anchorena			
4	23	D	1	None			
5	23	B	1	Juana Nazar			
<b>Nota:</b>							
None es la constante nula. Si aparece <b>None</b> en algún campo de un DataFrame es porque ese campo está vacío, no tiene contenido (equivale a <b>NaN</b> ).							
¿Qué contendrá <b>vue1</b> después de la siguiente operación?							
vue1=vue [ (vue [ 'fila' ]<20) & (vue [ 'asiento' ] != 'C' ) ]							
1	0	5	A	4	tripulación	1	
	1	16	A	2	disponible		
	2	16	B	2	Marcelo Uriondo		
	3	16	C	2	Delsy Anchorena		
	4	23	D	1	None		
	5	23	B	1	Juana Nazar		
2	0	5	A	4	tripulación	X	2
	1	16	A	2	None		
	2	16	B	2	Marcelo Uriondo		
3	5	23	B	1	Juana Nazar		3
4	1	2	16				4
	2	2	16				
	3	2	16				

0807 – 2 Ptos			
¿Cuáles modos de apertura deben emplearse con los archivos <b>datos.txt</b> y <b>otro.txt</b> en el siguiente programa para que no salte error?			
<pre>arch=open('datos.txt',...) txt='mi mama me mima' arch.write(txt+'\n') arch.close()  arch=open('otro.txt',...) arch.write('Fin\n') arch.close()</pre>			
1	open() de datos.txt 'a' open() de otro.txt 'w'	X	1
2	open() de datos.txt 'r' open() de otro.txt 'r+'		2
3	open() de datos.txt 'r' open() de otro.txt 'a'		3
4	open() de datos.txt 'w' open() de otro.txt 'r'		4

0907 – 2 Ptos

Para el DataFrame **vue** de pandas, que contiene:

	fila	asiento	zona	pax
0	5	A	4	tripulación
1	16	A	2	None
2	16	B	2	Marcelo Uriondo
3	16	C	2	Delsy Anchorena
4	23	D	1	None
5	23	B	1	Juana Nazar

**Nota:**  
**None** es la constante nula. Si aparece **None** en algún campo de un DataFrame es porque ese campo está vacío, no tiene contenido (equivale a **NaN**).

¿Qué operación produce el siguiente resultado?

asiento	
A	2
B	1
C	2
D	1

1	<code>vue.head(4)</code>		1
2	<code>vue[['pax']]</code>		2
3	<code>vue.groupby('asiento')['zona'].min()</code>	X	3
4	<code>vue['fila'].max()</code>		4

1007 – 2 Ptos

En el siguiente programa:

```
# por cada botón la clave es el código, los datos son color, precio
# unitario y tamaño en cm
botones={'MET 002':['plata',350.0,1],'MET 001':['dorado',475.6,1.5],
          'ACR 102':['rojo',210,0.8],'PVC 233 ':['azul',650,2]}
boton=input('Cgo botón: ').upper()
if boton in botones:
    ..... # línea a completar

    precio=round(botones[boton][1],2)
    print('El nuevo precio unitario de',boton,
          'con 10% de aumento es $',precio)
```

Que incrementa en un 10% el precio unitario de un artículo botón en una mercería. ¿Cuál debería ser la línea faltante en el código?

La salida final, para un ingreso **met 002** debería ser:

El nuevo precio unitario de MET 002 con 10% de aumento es \$ 385.0

1	<code>botones[boton][1]*=1.1</code>	X	1
2	<code>botones[boton]=1</code>		2
3	<code>boton= 350*10/100+350</code>		3
4	<code>botones[boton][2]-=1</code>		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