

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	DOCENTE (nombre y apellido):
TEL:	
AULA:	

Duración del examen: 1:30h. Completar con **letra clara, mayúscula e imprenta**. El examen consta de 13 preguntas de opción múltiple. Cada pregunta tiene una y sólo una respuesta correcta.

Las respuestas deben completarse en la siguiente matriz:

Opción	EJ. 1	EJ. 2	EJ. 3	EJ. 4	EJ. 5	EJ. 6	EJ. 7	EJ. 8	EJ. 9	EJ. 10	EJ. 11	EJ. 12	EJ. 13
1			X								X		
2		X			X					X			
3	X			X			X	X	X				
4						X						X	X

¡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 la de la matriz.

Ejercicio 0103

Dado el siguiente fragmento de código:

```
if ("a" in nombre and "i" not in nombre):
    print(1)
elif ("a" in nombre or "i" in nombre):
    print(2)
else:
    print(3)
if not (len(nombre) == 5):
    print(4)
```

¿Qué valor tiene que tener la variable nombre para que se imprima por pantalla exactamente un 2 y un 4?

1.	"maria"		1.
2.	"marlen"		2.
3.	"fito"	X	3.
4.	Ninguna de las opciones cumple con lo pedido		4.

Ejercicio 0203

Dado el siguiente fragmento de código:

```
i=1
while i<=3:
    i+=1
    print('vuelta')

for j in range(2,4):
    print('vuelta')

while i<2:
    i+=1
    print('vuelta')
```

¿Cuántas veces se imprime la palabra “vuelta”?

1.	4 (cuatro)		1.
2.	5 (cinco)	X	2.
3.	6 (seis)		3.
4.	7 (siete)		4.

Ejercicio 0303			
<p>Se quiere hacer un programa le pida al usuario ingresar su edad hasta que ingrese una edad válida. Para que una edad sea considerada válida debe cumplir con las siguientes condiciones:</p> <ul style="list-style-type: none">- Debe ser un entero.- Debe ser mayor o igual a 0 y menor o igual a 120. <p>¿Cuál de las siguientes versiones NO cumple el funcionamiento pedido?</p>			
1.	<pre>edad=int(input('Edad: ')) while edad<0 or edad>120: try: if edad<0 or edad>120: print('Seleccionar la edad entre 0 y 120') edad=int(input('Edad: ')) except ValueError: print('Sólo números enteros')</pre>	X	1.
2.	<pre>edad_valida=False while not edad_valida: try: edad=int(input('Edad: ')) if edad>=0 and edad<=120: edad_valida=True else: print('Seleccionar la edad entre 0 y 120') except ValueError: print('Sólo números enteros')</pre>		2.
3.	<pre>edad=-1 while edad<0 or edad>120: try: edad=int(input('Edad: ')) if edad<0 or edad>120: print('Seleccionar la edad entre 0 y 120') except ValueError: print('Sólo números enteros')</pre>		3.
4.	<pre>edad_valida=False while not edad_valida: try: edad=int(input('Edad: ')) if edad in range(0, 121): edad_valida=True else: print('Seleccionar la edad entre 0 y 120') except ValueError: print('Sólo números enteros')</pre>		4.

Ejercicio 0403			
Dadas las siguientes variables: <code>lista1 = [8, 5, 10]</code> <code>lista2 = [21, 42, 9]</code> <code>lista3 = []</code> ¿Cuál de los siguientes fragmentos de código deja a la lista3 con más de 3 elementos luego de ejecutarse?			
1.	<code>for i in range(3): if lista1[i] > 10: lista3.append(lista1[i]) elif lista2[i] > 10: lista3.append(lista2[i])</code>		1.
2.	<code>for i in lista1: if i >= 10: lista3.append(i) for i in lista2: if i <= 10: lista3.append(i)</code>		2.
3.	<code>for i in lista1: if i < 10: lista3.append(i)</code> <code>lista3 += lista2</code> <code>lista3.remove(8)</code>	X	3.
4.	<code>for i in lista2: if i < 10: lista3.append(i)</code> <code>lista3 += lista1</code> <code>lista3.remove(8)</code>		4.

Ejercicio 0503			
Dado el siguiente programa: <pre>def elige(dicci): print('Seleccioná') for op in dicci: print(op, '.', dicci[op]) seleccion = input() if seleccion not in dicci: return 0 return seleccion menu = {'1': 'Pasta', '2': 'Pollo', '3': 'Carne', '4': 'Veggie'} opc = elige(menu) print('Elegiste', menu[opc])</pre>			
¿Cuál de las siguientes afirmaciones es verdadera?			
1.	Se imprime 1 . Pasta 2 . Pollo 3 . Carne 4 . Veggie y permite que el usuario seleccione sólo opciones válidas. Luego devuelve la opción al programa principal.		1.
2.	Se imprime 1 . Pasta 2 . Pollo 3 . Carne 4 . Veggie y permite que el usuario seleccione cualquier opción. Luego la devuelve al programa principal. Si la opción no existe, se arroja un error.	X	2.
3.	Se imprime 1 . Pasta 2 . Pollo 3 . Carne 4 . Veggie y permite que el usuario seleccione cualquier opción. Luego devuelve la opción al programa principal. Si la opción no existe, se imprime 'Elegiste Pasta'.		3.
4.	Ninguna de las afirmaciones es verdadera		4.

Hubo una corrección en el punto 5 durante el examen. Había menos de 10 personas entre todas las aulas que habían entregado. A esas personas se les tomará el punto de forma especial y ya fueron identificados.

Ejercicio 0603			
<p>Dado el siguiente programa:</p> <pre>aulas = { 203: ['Matias', 'Tomás', 'Andrea'], 13: ['Bautista', 'Ana'], 116: ['Zoe', 'Miguel'] }</pre> <p># Acá falta un fragmento de código</p> <pre>for aula in aulas: print(aulas[aula][-1])</pre> <p>Cual de los siguientes fragmentos de código se puede poner en lugar del comentario para que por pantalla se imprima:</p> <p>Tomás Ana Miguel</p>			
1.	<pre>for aula in aulas: if aula>100: aulas[aula].sort(reverse=True)</pre>		1.
2.	<pre>for aula in aulas: if aula<100: aulas[aula].sort()</pre>		2.
3.	<pre>for aula in aulas: aulas[aula] = ['Tomás', 'Ana', 'Miguel']</pre>		3.
4.	Ninguna de las opciones cumple con lo pedido	X	4.

Ejercicio 0703			
<p>Dado el siguiente programa</p> <pre>lista_1 = [4, 3, 1, 2, 5] lista_2 = [10, -15, 5, 20, 15] suma = 0 for i in range(x, 4): j = lista_1[i] suma += lista_2[j] print(suma)</pre> <p>¿Cuál tiene que ser el valor de x para que se imprima un número menor a 10 ?</p>			
1.	0		1.
2.	1		2.
3.	2	X	3.
4.	Ninguna de las opciones es correcta		4.

Ejercicio 0803 Una empresa de videojuegos cuenta con un archivo que contiene por cada fila: código de uno de sus juegos, código de jugador, nivel alcanzado. El path del archivo está guardado en la variable file_name. Por ejemplo: 567994;29345;Platinado 567994;29340;Platinado 587721;29346;Completo 567990;29346;Platinado Se desea ver en pantalla un listado de códigos de juegos, sin repeticiones, que hayan sido platinados al menos una vez. ¿Cuál de los siguientes programas cumple con lo pedido ?			
1.	<pre>def platino(line): return "Platinado" in line file = open(file_name, "r") lineas = file.readlines() file.close() platinados = {} for linea in lineas: if platino(linea): platinados[linea.split(";")[0]] = True else: platinados[linea.split(";")[0]] = False for jugador in platinados: print(jugador)</pre>		1.
2.	<pre>def platino(line): return "Platinado" in line file = open(file_name, "r") lineas = file.readlines() file.close() platinados = [] for linea in lineas: if platino(linea): platinados.append(linea.split(";")[0]) for jugador in platinados: print(jugador)</pre>		2.
3.	<pre>def platino(line): return "Platinado" in line file = open(file_name, "r") lineas = file.readlines() file.close() platinados = [] for linea in lineas: if platino(linea) and linea.split(";")[0] not in platinados: platinados.append(linea.split(";")[0]) for jugador in platinados: print(jugador)</pre>	X	3.
4.	Ninguna de las opciones hace lo solicitado.		4.

Ejercicio 0903

Se tiene el siguiente DataFrame:

	nombre	artista	año	género	playlist	cant_reproducciones
0	Living la vida loca	Ricky Martin	1999	pop	1	5.0
1	La bestia Pop	PR y RR	1985	rock	2	NaN
2	Automático	María Becerra	2022	reggaetón	1	3.0
3	Stay	Rihanna	2012	pop	3	NaN
4	De música ligera	Soda Stereo	1990	rock	2	10.0

Si se quiere cambiar la columna 'playlist' para que, en vez de tener números, contenga los títulos:

- "Movidos" para la playlist 1
- "Rock nacional" para la 2
- "Románticos" para la 3.

¿Cuál es la forma correcta de hacer este cambio?

1.	df[df['playlist'].map({1: 'Movidos', 2: 'Rock nacional', 3: 'Románticos'})]		1.
2.	if (df['playlist'] == 1): df['playlist'] = "Movidos" elif (df['playlist'] == 2): df['playlist'] = "Rock nacional" else: df['playlist'] = "Románticos"		2.
3.	df['playlist'] = df['playlist'].map({1: 'Movidos', 2: 'Rock nacional', 3: 'Románticos'})	X	3.
4.	df['playlist'] = df['playlist'].apply({1: 'Movidos', 2: 'Rock nacional', 3: 'Románticos'})		4.

Ejercicio 1003			
Usando el mismo Dataframe del ejercicio 9, se quiere obtener el promedio de reproducciones de todas las canciones.			
¿Cuál de las siguientes opciones cumple con lo pedido?			
1.	df[df['cant_reproducciones']].mean()		1.
2.	df['cant_reproducciones'].mean()	X	2.
3.	df[df['cant_reproducciones'] != NaN].mean()		3.
4.	df['cant_reproducciones'].count().mean()		4.

Ejercicio 1103			
Considerando el DataFrame del punto 9 y el siguiente código:			
df['e'] = df['cant_reproducciones'].notnull()			
¿Cuál de las siguientes afirmaciones es correcta?			
1.	El código agrega la columna 'e' al dataframe original. Los valores de la columna van a ser True si la canción tiene cant_reproducciones, False si no tiene.	X	1.
2.	El código agrega la columna 'e' al dataframe original. Los valores de la columna van a ser False si la canción tiene cant_reproducciones, True si no tiene.		2.
3.	El código no agrega una columna al dataframe original. Sólo muestra una columna con los valores True (si la canción tiene cant_reproducciones) o False (si no tiene).		3.
4.	El código no agrega una columna al dataframe original. Sólo muestra una columna con los valores False (si la canción tiene cant_reproducciones) o True (si no tiene).		4.

Ejercicio 1203			
¿Cuál de estas afirmaciones es incorrecta?			
1.	La función <code>pie</code> recibe un arreglo de números. Puede o no recibir también un arreglo con labels para mostrar qué simboliza cada parte mostrada en el gráfico.		1.
2.	La función <code>scatter</code> recibe 2 arreglos que tienen que tener el mismo tamaño. Los arreglos representan una serie de puntos en el plano cartesiano.		2.
3.	La función <code>scatter</code> es muy parecida a la función <code>plot</code> , sólo que <code>scatter</code> no une los puntos con líneas.		3.
4.	Para la función <code>plot</code> , a menor cantidad de puntos, nuestra línea va a ser más suave.	X	4.

Ejercicio 1303			
<p>Se tiene el siguiente programa</p> <pre>def solo_voc(x): txt='' for char in x.lower(): if char in 'aeiouáéíóú': txt+=char return txt def pocas(x): return len(x)<3 a=['Ana', 'Eugenia', 'Milagros', 'Ema', 'Inés'] lista1=list(map(solo_voc,a)) lista2=list(filter(pocas,lista1))</pre>			
¿Qué contenido tendrá lista2 al ejecutarlo?			
1.	['ana', 'ema', 'inés']		1.
2.	['aa', 'eueia', 'iao', 'ea', 'ié']		2.
3.	['n', 'm', 'ns']		3.
4.	['aa', 'ea', 'ié']	X	4.