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 10 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
1										
2										
3										
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 la de la matriz.

Ejercicio 0101 - 1 punto

Para el siguiente DataFrame *paises*:

```
   idioma      país       continente   pobEnMillCopasFIFA
0  Español     España     Europa       47.000         1.0
1  Español     Bolivia    América      12.000         NaN
2  castellano  Argentina  América      45.000         3.0
3  Inglés      EEUU       América     331.000         NaN
4  Portugués   Brasil     América     214.000         5.0
5  Inglés      Australia  Oceanía      25.000         NaN
6  Holandés    Aruba      América       0.106         NaN
```

¿Qué instrucción produce la siguiente salida?

```
   idioma   país        continente  pobEnMill   CopasFIFA
1  Español  Bolivia     América      12.000     NaN
3  Inglés   EEUU        América     331.000     NaN
5  Inglés   Australia   Oceanía      25.000     NaN
6  Holandés Aruba       América       0.106     NaN
```

1	`paises.sort_values(['CopasFIFA','pobEnMill'])`		1
2	`paises[paises['pobEnMill']<100]`		2
3	`paises.groupby(['continente']).sum()`		3
4	`paises[paises['CopasFIFA'].isnull()]`	X	4

Ejercicio 0201 - 1 punto

¿Qué programa imprime más asteriscos?

1	`for letra in 'cielo':`
	`  i=1`
	`  while i < 4:`
	`  print('*')`
	`  i+=1`
	`  while i>2:`
	`  print('*')`
	`  i-=2`
		X	**1**

2	`for letra in 'cielito':`
	`  i=1`
	`  while i < 4:`
	`  print('*')`
	`  i+=3`
	`  print('*')`
	`for i in range(2,4,2):`
	`  print('*')`
			2

3	`for caracter in '* * *':`
	`  i=1`
	`  while i<=3:`
	`  print(caracter)`
	`  i+=2`
	`for i in range(2):`
	`  print('*')`
			3

4	`gira=False`
	`while gira:`
	`  i=1`
	`  while i<=6:`
	`  print('*')`
	`  i+=1`
	`for i in range(6):`
	`  print('*')`
			4

Ejercicio 0301 - 1 punto

¿Qué versión de *cuenta* funcionará correctamente en este programa?

```
def cuenta(…):
  -

nombres=['Ileana','Juan','Inés','EMILIANO']
vocales=[]
for nom in nombres:
  vocales.append(cuenta(nom))
maxVoc=max(vocales)
pos=vocales.index(maxVoc)
print('El nombre con más cantidad de vocales es:',nombres[pos])
```

Nota:
El método *index*() devuelve la posición donde se encuentra el elemento buscado.
Ej:

```
a=[1,0,-1]
a.index(0) -> 1
```

1	`def cuenta(t):`
	`  t=t.lower()`
	`  cont=0`
	`  for car in t:`
	`  if car in 'aeiouáéíóú':`
	`  cont+=1`
	`  return cont`
		X	**1**

2	`def cuenta(t,voc):`
	`  cont=0`
	`  for car in t:`
	`  if car in voc:`
	`  cont+=1`
	`  return cont`
			2

3	`def cuenta(t):`
	`  for car in t:`
	`  cont=0`
	`  if car in 'áéíóúaeiou':`
	`  cont+=1`
	`  return cont`
			3

4	`def cuenta():`
	`  for car in t:`
	`  if car in 'áéíóúaeiou':`
	`  cont=1`
	`  return cont`
			4

Ejercicio 0401 - 1 punto

¿Qué muestra el siguiente programa?

```
lista=[1,3,1,5,1,2]
while lista.count(1)>0:
  lista.remove(1)
for i in range(len(lista)):
  lista[i]*=10
print(lista)
```

1	`[10, 30, 10, 50, 10, 20]`		**1**
2	`[30, 50, 20]`	**X**	**2**
3	`[1, 3, 1, 5, 1, 2]`		**3**
4	`[3, 5, 2]`		**4**

Ejercicio 0501 - 1 punto

¿Qué versión de la función *numero* funcionará correctamente en el siguiente programa para obtener un ingreso por teclado de un número entre 11.0 y 22.0 inclusive? La función debe insistir hasta que el número ingresado esté efectivamente dentro del intervalo solicitado.

```
def numero(…):
  -
  -
  -

a=numero(11,22,'Ingresá un número entre 11 y 22 inclusive')
b=numero(11,22,'Ingresá un número entre 11 y 22 inclusive')
print(a+b)
```

1	`def numero():`
	`  while False:`
	`  try:`
	`  num=float(input(' '))`
	`  if num>=0 and num<=100:`
	`  return num`
	`  else:`
	`  print('Fuera del intervalo válido')`
	`  except:`
	`  print('Debe ser un número')`
			1

2	`def numero(cartel,hasta,desde):`
	`  while True:`
	`  try:`
	`  num=float(input(cartel+' '))`
	`  if num>=0 and num<=100:`
	`  return num`
	`  else:`
	`  print('Fuera del intervalo válido')`
	`  except:`
	`  print('Debe ser un número')`
	`  return 0`
			2

3	`def numero(desde,hasta,cartel):`
	`  while True:`
	`  try:`
	`  num=float(input(cartel+' '))`
	`  if num>=desde and num<=hasta:`
	`  return num`
	`  else:`
	`  print('Fuera del intervalo válido')`
	`  except:`
	`  print('Debe ser un número')`
		X	3

4	`def numero(desde,hasta,cartel):`
	`  num=0`
	`  try:`
	`  num=float(input(cartel+' '))`
	`  if num<desde or num>hasta:`
	`  print('Fuera del intervalo válido')`
	`  except:`
	`  print('Debe ser un número')`
	`  return num`
			4

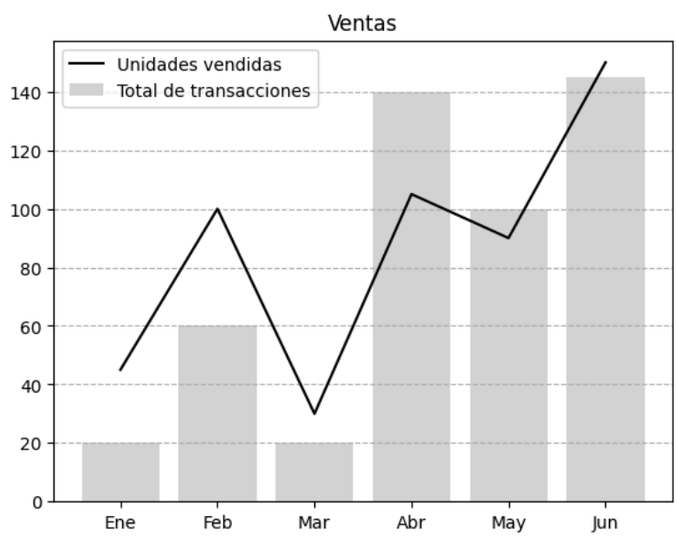
Ejercicio 0601 - 1 punto

¿Cuál invocación de *open()* funcionará adecuadamente para el siguiente programa que trabaja sobre el archivo *verbos.txt*?

```
arch=...
lineas=arch.readlines()
nuevaLin='\n'
for i in range(2):
  verb=input('Ingresá un infinitivo: ')
  term=verb[len(verb)-2:]
  while term.lower() not in ('ar','er','ir'):
    verb=input('Ingresá un infinitivo: ')
    term=verb[len(verb)-2:]
  nuevaLin=nuevaLin+verb+' '
arch.write(nuevaLin)
arch.close()
```

1	`open('verbos.txt','r+',encoding='utf-8')`	X	1
2	`open('w','verbos.txt','encoding'='utf-8')`		2
3	`open(verbos,encoding='a')`		3
4	`arch.open('verbos.txt','r',encoding='utf-8')`		4

Ejercicio 0701 - 1 punto



¿Cuál de las siguientes líneas de código SÍ SE CORRESPONDE con la figura?

1	`ax.bar(transacciones, dias, label = 'Total de transacciones', color= "lightgrey")`	X	1
2	`ax.bar(dias, unidades, label = 'Unidades vendidas', color= "lightgrey")`	X	2
3	`ax.plot(dias, transacciones, label = 'Total de transacciones', color="black")`	X	3
4	`ax.grid(axis = 'y', linestyle = 'dashed')`	X	4

Dado que en algunas copias de los exámenes los gráficos no se veían con claridad, se toman como correctas todas las opciones

Ejercicio 0801 - 2 puntos

¿Con qué contenido acabará la lista *nombres* cuando se ejecute el siguiente programa?

```
clave='samuel'
nombres=[12,17,21,3]
dicci={'s':'Ana','p':'Axel','m':'Teo','e':'José','i':'Juan',
     'o':'Lisa','u':'Inés','a':'Luis','l':'Ema'}
for i in range(len(nombres)):
  letra=clave[nombres[i]%len(clave)]
  nombres[i]=dicci[letra]
```

1	['Ana', 'Lisa', 'José', 'José']		1
2	[0, 5, 3, 3]		2
3	['Ana', 'Ema', 'Inés', 'Inés']	X	3
4	['Ana', 'Axel', 'Teo', 'José', 'Juan', 'Lisa', 'Inés', 'Luis', 'Ema']		4

Ejercicio 0901 - 2 puntos

¿Qué muestra el siguiente programa?

```
def cambia(nom):
  return nom.upper()

def selecciona(nom):
  return nom.count('A')>1

original=['Ana','LUCÍA','Alejandra','ALICIA','AntoNELLA']
sele=list(filter(selecciona,original))
final=list(map(cambia,sele))
print(final)
```

1	['Ana','Alejandra','ALICIA','AntoNELLA']		1
2	['ALICIA', 'ANTONELLA']	X	2
3	['LUCÍA']		3
4	['ANA','LUCÍA','ALEJANDRA','ALICIA','ANTONELLA']		4

Ejercicio 1001 - 2 puntos

Dado el siguiente condicional:

```
if a>30 and b!=abs(b):
  if c:
    print('Primavera')
  else:
    print('Invierno')
elif c and a%3==0:
  print('Otoño')
else:
  print('Verano')
```

Qué valores de ***a***,***b*** y ***c*** permiten que se imprima: **Otoño** ?

1	a=33, b=-45.6, c=True		1
2	a=33, b=45, c=False		2
3	a=52, b=-10, c=False		3
4	a=12, b=0, c=True	X	4