

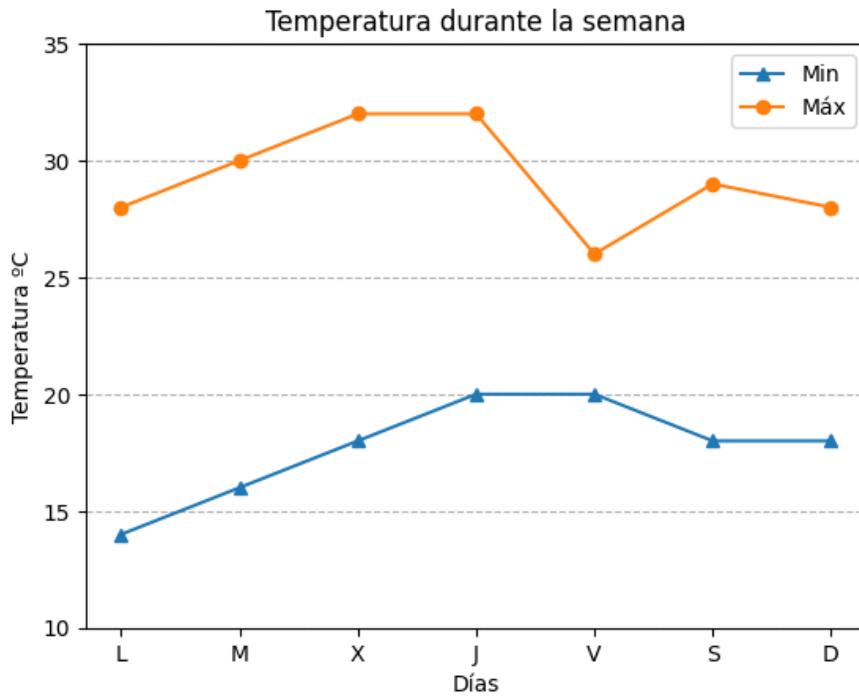




<b>Ejercicio 0501 - 1 punto</b>			
<p>¿Qué versión de la función <i>natural</i> funcionará correctamente en el siguiente programa para obtener un ingreso por teclado de un número natural (entero y mayor que 0)? La función debe insistir hasta que el número ingresado sea efectivamente un natural.</p> <pre>def natural(...):     -     -     -  a=natural() b=natural() print(a+b)</pre>			
<b>1</b>	<pre>def natural(num):     while True:         try:             num=int(input('Número natural: '))             if num&lt;=0:                 print('Ingrese un número natural')         except ValueError:             print('Ingrese un número natural')     return num</pre>		<b>1</b>
<b>2</b>	<pre>def natural():     pide=False     while pide:         try:             num=int(input('Número natural: '))             if num&gt;0:                 pide=False             else:                 pide=True                 print('Ingrese un número natural')         except ValueError:             print('Ingrese un número natural')     return num</pre>		<b>2</b>
<b>3</b>	<pre>def natural():     pide=True     while pide:         try:             num=int(input('Número natural: '))             pide=False             if num&lt;=0:                 print('Ingrese un número natural')         except ValueError:             print('Ingrese un número natural')             pide=False     return num</pre>		<b>3</b>
<b>4</b>	<pre>def natural():     while True:         try:             num=int(input('Número natural: '))             if num&gt;0:                 return num             else:                 print('Ingrese un número natural')         except ValueError:             print('Ingrese un número natural')</pre>	<b>X</b>	<b>4</b>

<b>Ejercicio 0601 - 1 punto</b>			
<p>¿Qué muestra el siguiente programa?</p> <pre>def letras(num):     let='aeiou'     return let[abs(num)%5]  lista=[2,22,6,10,-8] nueva=list(map(letras,lista)) print(nueva)</pre>			
<b>1</b>	[ ]		<b>1</b>
<b>2</b>	['i', 'i', 'e', 'a', 'o']	<b>X</b>	<b>2</b>
<b>3</b>	['O', 'A', 'E', 'I']		<b>3</b>
<b>4</b>	['A', 'E', 'I', 'O', '']		<b>4</b>

**Ejercicio 0701 - 1 punto**



¿Cuál de las siguientes líneas de código NO SE CORRESPONDE con la figura?

1	ax.plot(dias, temperaturas_min, marker = '^', label = 'Min')		1
	ax.plot(dias, temperaturas_max, marker = 'o', label = 'Máx')		
2	ax.grid(axis = 'y', linestyle = 'dashed')		2
3	ax.set_title("Temperatura durante la semana")		3
4	ax.set_xlim(10, 35)	X	4

**Ejercicio 0801 - 2 puntos**

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

```
def elige(...):
    -
    -
    -

gustos={'f': 'Frutilla', 'v': 'Vainilla',
        'd': 'Dulce de leche', 'c': 'Chocolate'}
salsas={'d': 'Ddl', 'c': 'Chocolate', 'f': 'Frutilla'}
copa=[]
print('Armá tu copa, 3 sabores')
for i in range(1,4):
    copa.append(elige(gustos, 'Sabor '+str(i)))
copa.append(elige(salsas, 'Qué salsa?'))
print('Mi copa es de',copa[0],',',copa[1],',y',copa[2],
      'con salsa de',copa[3])
```

1	def elige(cartel,dicci): print('Elegí tu opción con la letra resaltada') print(cartel) for opc in range(len(dicci)): print(dicci[opc]) sel=input().lower() while sel not in dicci: sel=input().lower()		1
2	def elige(dicci): print('Elegí tu opción con la letra resaltada') print('cartel') for opc in dicci: print(opc) sel=input().lower() while sel in dicci: sel=input().lower()		2
3	def elige(dicci, cartel): print('Elegí tu opción con la letra resaltada') print(cartel) for opc in dicci: print(dicci[opc]) sel=input().lower()	X	3

	<pre>while sel not in dicci:     sel=input().lower() return dicci[sel]</pre>		
4	<pre>def elige(dicci, cartel):     print('Elegí tu opción con la letra resaltada')     print(cartel)     sel=input().lower()     while sel not in dicci:         sel=input().lower()     return dicci['d']</pre>		4

Ejercicio 0901 - 2 puntos			
<p>¿ Qué valor para el argumento <i>modo</i> debería pasársele a la función <i>abrir</i> (en la primera y en la segunda invocación) para permitir que el programa guarde en mayúsculas las palabras del archivo <i>ar.txt</i>?</p> <pre>def abrir(arch, modo):     archivo=open(arch, modo, encoding='utf-8')     return archivo  verbos=abrir('ar.txt', ...) #Primera lineas=verbos.readlines() verbos.close()  verbos=abrir('ar.txt', ...) #Segunda for lin in lineas:     verbos.write(lin.upper()) verbos.close()</pre>			
1	Primera: 'r' Segunda: 'w'	X	1
2	Primera: 'r+' Segunda: 'a'	X	2
3	Primera: 'w+' Segunda: 'w'		3
4	Primera: 'a' Segunda: 'r+'		4

Dado que el enunciado es ambiguo y se puede interpretar como agregar o reemplazar, las opciones marcadas son las correctas.

Ejercicio 1001 - 2 puntos			
<p>Para el siguiente DataFrame <i>arts</i>:</p> <pre>      Cgo Desc          Pr Unit  Exist 0  LIB001 Lápiz negro HB      450.0  55.0 1  PAP112 Res Ledesma 70gr AA  2700.0  NaN 2  PAP099 Blck cuad Rivadavia A4958.5  2.0 3  MAP070 Planisferio nro 5     210.0  13.0 4  LIB015 Marc negro Sylvapen   675.0  11.0 5  LIB118 Regl Pizzini 30cm     387.0  NaN</pre> <p>¿Qué instrucción produce la siguiente salida?</p> <pre>      Cgo Desc          Pr Unit  Exist 1  PAP112 Res Ledesma 70gr AA  2700.0  NaN 5  LIB118 Regl Pizzini 30cm     387.0  NaN</pre>			
1	arts.head(2)		1
2	arts[arts['Exist'].isnull()]	X	2
3	arts[(arts['Pr Unit']<900)&(arts['Exist']>10)]		3
4	arts.describe()		4