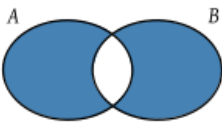
 PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA Facultad de Ingeniería y Ciencias Agrarias	Calificación	20 / 02 / 2018	FINAL	TEMA 1	
	LEGAJO:		INFORMÁTICA GENERAL		Hojas:
	APELLIDO:		NOMBRE:		
Para aprobar PARTE I y PARTE II deberá obtener al menos 27 puntos (sumando ambas partes).				[PARTE I Y II DE III]	

PARTE I - Completar programa (30 puntos completo. 15 puntos cada uno)

1 - Completar las secciones indicadas con “\_\_\_” con una línea de código correspondiente o secciones indicadas con “\_(\*)\_” con las líneas de código necesarias . El resto de las líneas de código del programa deben quedar sin modificar, y en el orden mostrado:

<p><b>1.1) -</b> Completar la función <b>ingresar</b> para que retorne el valor validado, ingresado por teclado. Sólo deberá aceptar el ingreso de números impares y múltiplos de 3, y además que no sean múltiplos de 7.</p> <pre>def ingresar ():      while _____:         print("Error")      return _____</pre>	<p><b>1.2) -</b> Completar la función <b>foo</b>, que recibe las listas <b>A</b> y <b>B</b> por parámetro. La función deberá retornar una nueva lista con la diferencia simétrica entre A y B. (Ver Figura)</p> <pre>def foo(A,B):     lst = []     _____     return lst</pre>
<p><u>Ejemplos: ingresar</u> <u>SI Debe</u> aceptar a: 3, 9, 15, 27 ..entre otros..</p>	<p><u>Ejemplos: ingresar NO Debe</u> aceptar a: 1, 2, 6, 5, 17, 18, 19, 21, 23 ..entre otros..</p>
<p>Diferencia simétrica entre <b>A</b> y <b>B</b> son aquellos elementos que se encuentran solo en <b>A</b> o bien sólo se encuentran en <b>B</b>, pero no se deben encontrar en ambos conjuntos (listas) a la vez.</p>	



PARTE II - Opción múltiple (32 puntos si todas correctas, 16 puntos si hay 3 correctas, 0 puntos otro caso)

2 - Indicar la opción correcta:

<p><b>2.1) -</b></p> <pre>def foo(miStr,num):     miStr[num]='x'     y=num-1     return miStr[num-y]</pre> <p>¿Qué retorna la función para <b>foo("abcd",len("abcd")-1)</b> ?</p> <p>a) axcd      b) abcx c) a          d) b e) Hay error en ejecución f) Ninguna de las anteriores</p>	<p><b>2.2) -</b></p> <pre>import random def aleatorio():     lst=[]     for i in range(0,5):         x=(random.randint(0,20)%5)*2         lst.append(x)     return lst</pre> <p>¿Cuál es la lista que pudo haber retornado la función <b>aleatorio()</b>?</p> <p>a) [0,1,2,3,4,5] b) [1,2,3,4,5] c) [1,2,3,4,5,6,7,8] d) [0,2,4,6,8] e) Ninguna de las anteriores</p>
<p><b>2.3) -</b></p> <pre>lst=[10,20,30,40,50] tam=len(lst) i=1 suma=0 while i&lt;=tam:     suma+=lst[i]     i+=1 print(suma)</pre> <p>¿Qué imprime al ejecutar el código ?</p> <p>a) 150      b) 140      c) 100      d) 90 e) Ninguna de las anteriores</p>	<p><b>2.4) -</b></p> <pre>def foo():     sum='0'     for x in "0123":         if not(x=='0'and x=='1')== (not x=='0' or not x=='1'):             sum=sum+x     return sum</pre> <p>¿Qué retorna al ejecutar el código de <b>foo()</b> ?</p> <p>a) 0          b) 00123      c) 0012 d) 0000      e) 00000 f) Hay error en ejecución g) Ninguna de las anteriores</p>