PENSAMIENTO COMPUTACIONAL (90) .UBAXXI

EXAMEN: PRIMER PARIAL – PRIMER CUATRIMESTRE 2025 **TEMA 7**

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	NOTA Y FIRMA DOCENTE (no rellenar)
TEL:	
AULA:	

Duración del examen: 1:30h.

- ✔ Escribir claramente el nombre en todas las páginas.
- ✔ El examen consta de 9 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 5 inclusive permiten acumular 1 punto (si son correctas), de la 6 a la 9 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**

Consignar con una X (cruz) la respuesta elegida en cada ejercicio en la siguiente matriz:

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Talón de Control para el Alumno (Tema 7)

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Ejercicio 1 – Tema 7			**1 Pto**
¿Cuál de los siguientes programas muestra por pantalla **caro**?			

A	`producto='birome'`
	`unidad='caja'`
	`precio=60`
	`if (producto in ('lápiz color','birome',`
	`'boligoma')):`
	`  if precio<100 and unidad=='caja' and unidad=='kg':`
	`  print('carito')`
	`  elif precio<100 and unidad=='unidad':`
	`  print('caro')`
	`elif producto in ('plasticola','pincel','hoja'):`
	`  print('barato')`
			A

B	`producto='birome'`
	`unidad='caja'`
	`precio=60`
	`if producto in ('block A4','etiqueta'):`
	`  if (precio>100 and (unidad=='caja' or`
	`unidad=='docena')):`
	`  print('carito')`
	`  elif precio<100 and unidad=='unidad':`
	`  print('maso')`
	`elif producto in ('plasticola','hoja','birome'):`
	`  if unidad in ('ciento','docena','bolsa'):`
	`  print('barato')`
	`  else:`
	`  print('caro')`
		X	**B**

C	`producto='birome'`
	`unidad='caja'`
	`precio=60`
	`if producto in ('birome','etiqueta','marcador'):`
	`  if (precio>100 and`
	`(unidad=='caja' or unidad=='docena')):`
	`  print('precio medio')`
	`  elif precio>100 and unidad=='unidad':`
	`  print('caro')`
	`print('barato')`
			C

D	`producto='birome'`
	`unidad='caja'`
	`precio=60`
	`if producto in ('lápiz color','marcador','birome'):`
	`  if (precio>100 or unidad=='caja'`
	`or unidad=='kg'):`
	`  print('precio medio')`
	`  elif precio>100 and unidad=='unidad':`
	`  print('caro')`
	`else:`
	`  print('barato')`
			D

Ejercicio 2 – Tema 7	1 Pto

¿Qué muestra el siguiente programa?

```
a=0
b=1
c=2
d=-2
cant=d-b*b+c*2/b+4
print('&'* int(cant))
```

Nota:
El operador * aplicado a una string y un número entero es repetición
Ej:
'hola'*2-> 'holahola'

A	3		**A**
B	`&&&&&`	**X**	**B**
C	`&&`		**C**
D	`&&&`		**D**

Ejercicio 3 – Tema 7	1 Pto

¿Con qué contenido acabará la lista **final** en el siguiente programa?

```
indicador=[1,2,3,4,5,6]
numeros=[1,2,3,4,5,6]
final=[0]*6
for i in range(len(numeros)):
    final[-(indicador[i])]=numeros[i]
```

Notas:
El operador * aplicado a una lista es repetición
Ej:
[1, 3]*2 -> [1, 3, 1, 3]

Se puede referenciar de atrás para adelante un elemento de una lista con un índice negativo
Ej:
a=[1,2,3,4]
a[-2] -> 3 y **a[-1] -> 4**

A	`[0, 0, 0, 6, 5, 4]`		**A**
B	`[6, 5, 4, 3, 2, 1]`	**X**	**B**
C	`[0, 0, 0]`		**C**
D	`[1, 2, 3, 4, 5, 6]`		**D**

Ejercicio 4 – Tema 7	**1 Pto**

¿Cuál programa carga **ordenadamente** en forma **descendente** las listas **enteros** y **fraccionarios** con los elementos de **numeros** que corresponda? En **enteros** debe quedar **[10, 2, -3]** y en **fraccionarios [0.25, -3.36]**

Notas:
El método ***reverse()*** invierte una lista
Ej:
lis=[-1,2,0] lis.reverse() lis-> [0,2,-1]

El método ***sort()*** ordena una lista
Ej:
lis=[6,-1,0] lis.sort() -> lis=[-1, 0, 6]

El método ***copy()*** genera una copia de la lista
Ej:
lis=[6,0]
lis1=lis.copy() -> hay dos listas que en ese momento tienen los mismos datos

El método ***remove()*** quita la primer ocurrencia de un elemento en una lista
Ej:
lis=[6,-1,0,5,6] lis.remove(6) -> lis=[-1, 0, 5, 6]

A

```
numeros=[-3.36,10,2,0.25,-3]
enteros=[]
fraccionarios=[]
for num in numeros:
    if num==int(num):
        fraccionarios.append(num)
    else:
        fraccionarios.append(num)
enteros.sort()
fraccionarios.reverse()
```

A

B

```
numeros=[-3.36,10,2,0.25,-3]
enteros=numeros.copy()
fraccionarios=numeros.copy()
for num in enteros:
    if num!=int(num):
        enteros.remove(num)
enteros.sort()
fraccionarios.sort()
```

B

C

```
numeros=[-3.36,10,2,0.25,-3]
enteros=[]
fraccionarios=[]
for num in numeros:
    if num==int(num):
        enteros.append(num)
    else:
        fraccionarios.append(num)
enteros.sort()
enteros.reverse()
fraccionarios.sort()
fraccionarios.reverse()
```

X

C

D

```
numeros=[-3.36,10,2,0.25,-3]
enteros=[]
fraccionarios=[]
for num in enteros:
    if num==int(num):
        numeros.append(num)
    else:
        fraccionarios.append(num)
numeros.reverse()
fraccionarios.reverse()
```

D

Ejercicio 5 – Tema 7 **1 Pto**

¿Qué devuelve la función **fun** en este programa?

```
def fun(t):
    vocales='AEIOUÁÉÍÓÚÜ'
    repes=False
    repetidas=''
    for letra in t:
        if  letra in vocales and t.count(letra)>0:
            repes=True
            if letra not in repetidas:
                repetidas+=letra
    return repetidas.lower()

txt='Estaba LA RANA cantando debajo del AGUA'
print(fun(txt))
```

Nota:
El método ***count()*** cuenta cuántas apariciones de su argumento hay en una string
Ej:
'hola, qué tal?'.count('a') -> 2

A	`'eau'`	**X**	**A**
B	`False`		**B**
C	`True`		**C**
D	`'EAAAAUA'`		**D**

Ejercicio 6 – Tema 7 **2 Ptos**

¿Qué muestra por pantalla el siguiente programa?

```
impares='13579'
for i in range(len(impares)):
  linea=''
  for j in range(len(impares)*2):
    if j%2!=0:
      linea+=impares[j//2]
    else:
      linea+=' '
  print(linea)
```

Notas:
El operador * aplicado a una string es repetición
Ej:
'hola'*3 -> 'holaholahola'

El operador // es el cociente entero
Ej:
7//2 -> 3

A	`1`
	` 2`
	`  3`
	`   4`
	`    5`
	`     6`
	`      7`
	`       8`
	`        9`
			A

B	` 1 3 5 7 9` ` 1 3 5 7 9` ` 1 3 5 7 9` ` 1 3 5 7 9` ` 1 3 5 7 9`	**X**	**B**
C	`123456789` `987654321`		**C**

D	`1`
	`1 3`
	`1 3 5`
	`1 3 5 7`
	`1 3 5 7 9`
			D

Ejercicio 7 – Tema 7			2 Ptos
¿Cuál programa muestra el siguiente texto como salida? **LAS ABEJAS VAALAN Y LAS OVEJAS BALAN** **Notas:** El método replace() permite cambiar todas las apariciones de una string por otra dentro de un texto **Ej:** **'llueve a las doce. Qué llueva, qué llueva!'.replace(‘llue’,’yu’)** **-> ''yuve a las doce. Qué yuva, qué yuva!'** Se puede seleccionar una porción de una string con **[desde:hasta]**, (recordá que **hasta** no entra) **Ej:** **'hola, qué tal?'[4:] -> ', qué tal?'**			

A	`frase='Las abejas vuelan y las ovejas balan'`
	`animal1='abejas'`
	`animal2='ovejas'`
	`frase1=frase[:18]`
	`frase2=frase[18:]`
	`frase1=frase1.replace(animal1,animal2)`
	`frase2=frase2.replace(animal2,animal1)`
	`frase1=frase1.replace('vuelan','no vuelan')`
	`frase2=frase2.replace('balan','no balan')`
	`frase=frase1+frase2.upper()`
	`print(frase)`
			A

B	`frase='Las abejas vuelan y las ovejas balan'`
	`animal1='abeja'`
	`animal2='oveja'`
	`frase=frase.replace(animal1,'ABEJA')`
	`frase=frase.replace(animal2,'OVEJA')`
	`for voc in 'EIOU':`
	`  frase=frase.replace(voc,'A')`
	`print(frase)`
			B

C	`frase='Las abejas vuelan y las ovejas balan'`
	`animal1='abejas'`
	`animal2='ovejas'`
	`frase=frase.replace(animal1,animal2)`
	`frase=frase.replace(animal2,animal1)`
	`frase=frase.replace('vuelan','no vuelan')`
	`frase.replace('balan','no balan')`
	`print(frase)`
			C

D	`frase='Las abejas vuelan y las ovejas balan'`
	`animal1='abeja'`
	`animal2='oveja'`
	`frase=frase.replace(animal1,'ABEJA')`
	`frase=frase.replace(animal2,'OVEJA')`
	`for voc in 'eiou':`
	`  frase=frase.replace(voc,'A')`
	`print(frase.upper())`
		X	**D**

Ejercicio 8 – Tema 7 | **2 Ptos**

¿Cuál es la versión correcta de la función **edita**? El programa debe mostrar
argumentar CAOS en cualquier situación no es COMPROMISO

```
def edita(…):
  -
  -
  -

txt='argumentar caos en cualquier situación no es compromiso'
resalta=['compromiso','éxito','objetivos','caos','ganancias']
print(edita(txt,resalta))
```

Notas:
El método ***split()*** genera una lista con las partes de una string separadas por blancos
Ej:
txt='hola qué tal a todos'
lis= txt.split() lis-> ['hola', 'qué', 'tal', 'a', 'todos']

El método ***join()*** genera una string con los caracteres que tiene como elementos una lista separándolos por el texto al inicio
Ej:
lis= ['h', 'o', 'l', 'a']
txt=''.join(lis)-> txt='h**o**l**a'**

A

```
def edita(t,palabras):
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in palabras:
            lista[i]=lista[i].lower()
```

A

B

```
def edita(t):
    t.lower()
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in palabras:
            lista[i]=lista[i]
    return ' '.join(palabras)
```

B

C

```
def edita(t,palabras):
    t=t.lower()
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in palabras:
            lista[i]=lista[i].upper()
    return ' '.join(lista)
```

X

C

D

```
def edita(t,palabras):
    t=t.lower()
    lista=t.split()
    for i in range(len(t)):
        if t[i] in palabras:
            t[i]=t[i].upper()
    t=' '.join(palabras)
    return t.upper()
```

D

Ejercicio 9 – Tema 7 **2 Ptos**

¿Cuál condición es candidata a ser equivalente a la siguiente?

(num not in ent and num not in fracc and
((num==int(num) and num<10)or(num!=int(num) and num<1)))

Atención!!
Te recomendamos que pruebes las condiciones con los siguientes valores:

```
ent=[-2,21,16]
fracc=[33.5,-0.21,12.05]
num=-2 y num=2.05
```

Nota:
Una condición (proposición) **p** es equivalente a otra **q**, si sus tablas de verdad coinciden. En otras palabras: si siempre que **p** da **V**, **q** da **V** y siempre que **p** da **F**, **q** da **F**

A	`((int(num)!=num or num==int(num) or num<=9) and (num not in fracc or num not in ent))`		**A**
B	`(int(num)!=num or num==int(num) or num not in fracc or num not in ent)`		**B**
C	`(((int(num)!=num and num<1)or (num==int(num) and num<=9)) and num not in fracc and num not in ent)`	**X**	**C**
D	`num in ent and num==int(num) and num<1`		**D**

PENSAMIENTO COMPUTACIONAL (90)

.UBAXXI

EXAMEN: PRIMER PARIAL – PRIMER CUATRIMESTRE 2025

TEMA 8

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	NOTA Y FIRMA DOCENTE (no rellenar)
TEL:	
AULA:	

Duración del examen: 1:30h.

- ✔ Escribir claramente el nombre en todas las páginas.
- ✔ El examen consta de 9 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 5 inclusive permiten acumular 1 punto (si son correctas), de la 6 a la 9 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**

Consignar con una X (cruz) la respuesta elegida en cada ejercicio en la siguiente matriz:

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Talón de Control para el Alumno (Tema 8)

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Ejercicio 1 – Tema 8			**1 Pto**
¿Cuál de los siguientes programas muestra por pantalla **barato**?			

A	`producto='birome'`
	`unidad='caja'`
	`precio=60`
	`if (producto in ('lápiz color','birome',`
	`'boligoma')):`
	`  if precio<100 and unidad=='caja' and unidad=='kg':`
	`  print('carito')`
	`  elif precio<100 and unidad=='unidad':`
	`  print('caro')`
	`elif producto in ('plasticola','pincel','hoja'):`
	`  print('barato')`
			A

B	`producto='birome'`
	`unidad='caja'`
	`precio=60`
	`if producto in ('block A4','etiqueta'):`
	`  if (precio>100 and (unidad=='caja' or`
	`unidad=='docena')):`
	`  print('carito')`
	`  elif precio<100 and unidad=='unidad':`
	`  print('maso')`
	`elif producto in ('plasticola','hoja','birome'):`
	`  if unidad in ('ciento','docena','bolsa'):`
	`  print('barato')`
	`  else:`
	`  print('caro')`
			B

C	`producto='birome'`
	`unidad='caja'`
	`precio=60`
	`if producto in ('birome','etiqueta','marcador'):`
	`  if (precio>100 and`
	`(unidad=='caja' or unidad=='docena')):`
	`  print('precio medio')`
	`  elif precio>100 and unidad=='unidad':`
	`  print('caro')`
	`print('barato')`
		X	**C**

D	`producto='birome'`
	`unidad='caja'`
	`precio=60`
	`if producto in ('lápiz color','marcador','birome'):`
	`  if (precio>100 or unidad=='caja'`
	`or unidad=='kg'):`
	`  print('precio medio')`
	`  elif precio>100 and unidad=='unidad':`
	`  print('caro')`
	`else:`
	`  print('barato')`
			D

Ejercicio 2 – Tema 8	**1 Pto**

¿Qué muestra el siguiente programa?

```
a=0
b=1
c=2
d=-2
cant=(d-b)*b+c*(2/b)+1
print('&'* int(cant))
```

Nota:
El operador * aplicado a una string y un número entero es repetición
Ej:
'hola'*2-> 'holahola'

A	3		**A**
B	&&&&&		**B**
C	&&	**X**	**C**
D	&&&		**D**

Ejercicio 3 – Tema 8	**1 Pto**

¿Con qué contenido acabará la lista **final** en el siguiente programa?

```
indicador=[1,2,3,1,2,3]
numeros=[1,2,3,4,5,6]
final=[0]*6
for i in range(len(numeros)):
    final[-(indicador[i])]=numeros[i]
```

Notas:
El operador * aplicado a una lista es repetición
Ej:
[1, 3]*2 -> [1, 3, 1, 3]

Se puede referenciar de atrás para adelante un elemento de una lista con un índice negativo
Ej:
a=[1,2,3,4]
a[-2] -> 3 y **a[-1] -> 4**

A	[0, 0, 0, 6, 5, 4]	**X**	**A**
B	[6, 5, 4, 3, 2, 1]		**B**
C	[0, 0, 0]		**C**
D	[1, 2, 3, 4, 5, 6]		**D**

Ejercicio 4 – Tema 8 | **1 Pto**

¿Cuál programa carga **ordenadamente** en forma **descendente** las listas **enteros** y **fraccionarios** con los elementos de **numeros** que corresponda? En **enteros** debe quedar **[10, 2, -3]** y en **fraccionarios [0.25, -3.36]**

Notas:
El método ***reverse()*** invierte una lista
Ej:
lis=[-1,2,0] lis.reverse() lis-> [0,2,-1]

El método ***sort()*** ordena una lista
Ej:
lis=[6,-1,0] lis.sort() -> lis=[-1, 0, 6]

El método ***copy()*** genera una copia de la lista
Ej:
lis=[6,0]
lis1=lis.copy() -> hay dos listas que en ese momento tienen los mismos datos

El método ***remove()*** quita la primer ocurrencia de un elemento en una lista
Ej:
lis=[6,-1,0,5,6] lis.remove(6) -> lis=[-1, 0, 5, 6]

A

```
numeros=[-3.36,10,2,0.25,-3]
enteros=[]
fraccionarios=[]
for num in numeros:
    if num==int(num):
        fraccionarios.append(num)
    else:
        fraccionarios.append(num)
enteros.sort()
fraccionarios.reverse()
```

A

B

```
numeros=[-3.36,10,2,0.25,-3]
enteros=numeros.copy()
fraccionarios=[]
for num in numeros:
    if num!=int(num):
        enteros.remove(num)
        fraccionarios.append(num)
fraccionarios.sort()
enteros.sort()
fraccionarios.reverse()
enteros.reverse()
```

X

B

C

```
numeros=[-3.36,10,2,0.25,-3]
enteros=numeros.copy()
fraccionarios=numeros.copy()
for num in enteros:
    if num!=int(num):
        enteros.remove(num)
enteros.sort()
fraccionarios.sort()
```

C

D

```
numeros=[-3.36,10,2,0.25,-3]
enteros=[]
fraccionarios=[]
for num in enteros:
    if num==int(num):
        numeros.append(num)
    else:
        fraccionarios.append(num)
numeros.reverse()
fraccionarios.reverse()
```

D

Ejercicio 5 – Tema 8 **1 Pto**

¿Qué devuelve la función **fun** en este programa?

```
def fun(t):
    vocales='AEIOUÁÉÍÓÚÜ'
    repes=False
    repetidas=''
    for letra in t:
        if  letra in vocales and t.count(letra)>0:
            repes=True
            if letra not in repetidas:
                repetidas+=letra
    return 'O' in repetidas.lower()

txt='Estaba LA RANA cantando debajo del AGUA'
print(fun(txt))
```

Nota:
El método ***count()*** cuenta cuántas apariciones de su argumento hay en una string
Ej:
'hola, qué tal?'.count('a') -> 2

A	`'eau'`		**A**
B	`False`	**X**	**B**
C	`True`		**C**
D	`'EAAAAUA'`		**D**

Ejercicio 6 – Tema 8 **2 Ptos**

¿Qué muestra por pantalla el siguiente programa?

```
impares='13579'
for i in range(len(impares)):
  linea=''
  for j in range(len(impares)*2):
    if j%2!=0:
      linea+=impares[j//2]
    else:
      linea+=' '
  if i%2==0:
    print(linea)
  else:
    print()
```

Notas:
El operador * aplicado a una string es repetición
Ej:
'hola'*3 -> 'holaholahola'

El operador // es el cociente entero
Ej:
7//2 -> 3

A	`1`
	` 2`
	`  3`
	`   4`
	`    5`
	`     6`
	`      7`
	`       8`
	`        9`
			A

B	` 1 3 5 7 9` ` 1 3 5 7 9` ` 1 3 5 7 9` ` 1 3 5 7 9` ` 1 3 5 7 9`		**B**
C	`123456789` `987654321`		**C**
D	` 1 3 5 7 9` ` 1 3 5 7 9` ` 1 3 5 7 9`	**X**	**D**

Ejercicio 7 – Tema 8			2 Ptos
¿Cuál programa muestra el siguiente texto como salida? **Las ovejas no vuelan Y LAS ABEJAS NO BALAN** **Notas:** El método replace() permite cambiar todas las apariciones de una string por otra dentro de un texto **Ej:** **'llueve a las doce. Qué llueva, qué llueva!'.replace(‘llue’,’yu’)** **-> ''yuve a las doce. Qué yuva, qué yuva!'** Se puede seleccionar una porción de una string con **[desde:hasta]**, (recordá que **hasta** no entra) **Ej:** **'hola, qué tal?'[4:] -> ', qué tal?'**			

A	`frase='Las abejas vuelan y las ovejas balan'`
	`animal1='abejas'`
	`animal2='ovejas'`
	`frase1=frase[:18]`
	`frase2=frase[18:]`
	`frase1=frase1.replace(animal1,animal2)`
	`frase2=frase2.replace(animal2,animal1)`
	`frase1=frase1.replace('vuelan','no vuelan')`
	`frase2=frase2.replace('balan','no balan')`
	`frase=frase1+frase2.upper()`
	`print(frase)`
		X	**A**

B	`frase='Las abejas vuelan y las ovejas balan'`
	`animal1='abeja'`
	`animal2='oveja'`
	`frase=frase.replace(animal1,'ABEJA')`
	`frase=frase.replace(animal2,'OVEJA')`
	`for voc in 'EIOU':`
	`  frase=frase.replace(voc,'A')`
	`print(frase)`
			B

C	`frase='Las abejas vuelan y las ovejas balan'`
	`animal1='abejas'`
	`animal2='ovejas'`
	`frase=frase.replace(animal1,animal2)`
	`frase=frase.replace(animal2,animal1)`
	`frase=frase.replace('vuelan','no vuelan')`
	`frase.replace('balan','no balan')`
	`print(frase)`
			C

D	`frase='Las abejas vuelan y las ovejas balan'`
	`animal1='abeja'`
	`animal2='oveja'`
	`frase=frase.replace(animal1,'ABEJA')`
	`frase=frase.replace(animal2,'OVEJA')`
	`for voc in 'eiou':`
	`  frase=frase.replace(voc,'A')`
	`print(frase.upper())`
			D

Ejercicio 8 – Tema 8 **2 Ptos**

¿Cuál es la versión correcta de la función **edita**? El programa debe mostrar
el ÉXITO depende de los OBJETIVOS y las GANANCIAS esperadas

```
def edita(…):
  -
  -
  -

txt='el éxito depende de los objetivos y las ganancias esperadas'
resalta=['compromiso','éxito','objetivos','caos','ganancias']
print(edita(txt,resalta))
```

Notas:
El método ***split()*** genera una lista con las partes de una string separadas por blancos
Ej:
txt='hola qué tal a todos'
lis= txt.split() lis-> ['hola', 'qué', 'tal', 'a', 'todos']

El método ***join()*** genera una string con los caracteres que tiene como elementos una lista separándolos por el texto al inicio
Ej:
lis= ['h', 'o', 'l', 'a']
txt=''.join(lis)-> txt='h**o**l**a'**

A

```
def edita(t,palabras):
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in palabras:
            lista[i]=lista[i].lower()
```

( ) **A**

B

```
def edita(t):
    t.lower()
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in palabras:
            lista[i]=lista[i]
    return ' '.join(palabras)
```

( ) **B**

C

```
def edita(t,palabras):
    t=t.lower()
    lista=t.split()
    for i in range(len(t)):
        if t[i] in palabras:
            t[i]=t[i].upper()
    t=' '.join(palabras)
    return t.upper()
```

( ) **C**

D

```
def edita(t,palabras):
    t=t.lower()
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in palabras:
            lista[i]=lista[i].upper()
    return ' '.join(lista)
```

(**X**) **D**

Ejercicio 9 – Tema 8	**2 Ptos**

¿Cuál condición es candidata a ser equivalente a la siguiente?

(num not in ent and num not in fracc and
((num==int(num) and num<10)or(num!=int(num) and num<1)))

Atención!!
Te recomendamos que pruebes las condiciones con los siguientes valores:

```
ent=[-2,21,16]
fracc=[33.5,-0.21,12.05]
num=-2 y num=2.05
```

Nota:
Una condición (proposición) **p** es equivalente a otra **q**, si sus tablas de verdad coinciden. En otras palabras: si siempre que **p** da **V**, **q** da **V** y siempre que **p** da **F**, **q** da **F**

A	`num in ent and num==int(num) and num<1`		**A**
B	`((num==int(num) and num<10 and (num not in ent and num not in fracc))or (num!=int(num) and num<1 and (num not in ent and num not in fracc)))`	**X**	**B**
C	`((int(num)!=num or num==int(num) or num<=9) and (num not in fracc or num not in ent))`		**C**
D	`(int(num)!=num or num==int(num) or num not in fracc or num not in ent)`		**D**

PENSAMIENTO COMPUTACIONAL (90)

.UBAXXI

EXAMEN: PRIMER PARIAL – PRIMER CUATRIMESTRE 2025

TEMA 6

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	NOTA Y FIRMA DOCENTE (no rellenar)
TEL:	
AULA:	

Duración del examen: 1:30h.

- ✔ Escribir claramente el nombre en todas las páginas.
- ✔ El examen consta de 9 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 5 inclusive permiten acumular 1 punto (si son correctas), de la 6 a la 9 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**

Consignar con una X (cruz) la respuesta elegida en cada ejercicio en la siguiente matriz:

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Talón de Control para el Alumno (Tema 6)

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Ejercicio 1 – Tema 6			**1 Pto**
¿ Cuál de los siguientes programas muestra por pantalla **nacional e importado**?			

A	`producto='heladera'`
	`proveedor='Lider s.a.'`
	`precio=750000`
	`if (producto =='freezer vertical' and`
	`  producto=='heladera' and`
	`  producto=='lavavajillas'):`
	`  if proveedor in ('lastrech','dami','electrtech'):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`print('solo nacional')`
			A

B	`producto='heladera'`
	`proveedor='Lider s.a.'`
	`precio=750000`
	`if (producto =='heladera' or`
	`  producto=='ventilador' or`
	`  producto=='lavarropas'):`
	`  if (proveedor in ('lastrech','dami',`
	`  'atlas')):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`elif (producto in ('freezer horizontal',`
	`  'freezer vertical',`
	`  'anafe')):`
	`  print('solo nacional')`
		X	**B**

C	`producto='heladera'`
	`proveedor='Lider s.a.'`
	`precio=750000`
	`if producto in ('heladera','freezer vertical'):`
	`  if (proveedor not in ('lastrech','dami',`
	`  'atlas')):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`elif (producto in ('anafe','lavavajillas',`
	`  'lavaropas')):`
	`  print('solo nacional')`
			C

D	`producto='heladera'`
	`proveedor='Lider s.a.'`
	`precio=750000`
	`if producto in ('horno','anafe'):`
	`  if proveedor!='Lider s.a.':`
	`  print('solo importado')`
	`  else:`
	`  print('nacional e importado')`
	`else:`
	`  print('solo nacional')`
			D

Ejercicio 2 – Tema 6			1 Pto

¿Qué muestra el siguiente programa?
 	
`a=0.5`
`b=-1`
`c=1.5`
`d=2`
`print((a+d*b)/c+b)`			

A	`0.4`		**A**
B	`2.5`		**B**
C	`0.0`		**C**
D	`-2.0`	**X**	**D**

Ejercicio 3 – Tema 6			1 Pto

¿Con qué contenido acabará la lista **final** en el siguiente programa?
 	
`indicador=[0,6,5,9,8,9]`
`numeros=[1,2,3,4,5,6]`
`final=[0]*10`
`for i in range(len(numeros)):`
`  final[indicador[i]]=numeros[i]`
 	
Nota:
El operador * aplicado a una lista es repetición
Ej:
[1, 3]*2 -> [1, 3, 1, 3]			

A	`[0, 0, 0, 0, 0, 0]`		**A**
B	`[1, 0, 0, 0, 0, 3, 2, 0, 5, 6]`	**X**	**B**
C	`[4, 0, 0, 0, 0, 6, 5, 0, 0, 0]`		**C**
D	`[6, 5, 4, 3, 2, 1]`		**D**

Ejercicio 4 – Tema 6			**1 Pto**
¿Cuál programa integra **ordenadamente** en la lista **numeros** tanto los pares como los impares? En **numeros** debe quedar **[3, 7, 11, 14, 15, 22]** **Notas:** El método ***reverse()*** invierte una lista **Ej:** **lis=[-1,2,0]** **lis.reverse() lis-> [0,2,-1]** El método ***sort()*** ordena una lista **Ej:** **lis=[6,-1,0]** **lis.sort() -> lis=[-1, 0, 6]** El método ***extend()*** agrega una lista a otra **Ej:** **lis1=[6,-1,5,6]** **lis2=[0,0]** **lis1.extend(lis2) -> lis1=[6, -1, 5, 6, 0, 0]** El método ***insert(posición,nuevo elemento)*** permite agregar un elemento en una lista en cualquier posición **Ej:** **lis=[1, 2, 3]** **lis.insert(1,8) -> lis=[1, 8, 2, 3]**			

A	`impares=[15,7,11,3]`
	`pares=[22,14]`
	`numeros=[]`
	`for i in range(len(pares)):`
	`  numeros.append(impares[i])`
	`  numeros.append(pares[i])`
			A

B	`impares=[15,7,11,3]`
	`pares=[22,14]`
	`numeros=impares`
	`for num in pares:`
	`  if num not in numeros:`
	`  numeros.insert(0,num)`
			B

C	`impares=[15,7,11,3]`
	`pares=[22,14]`
	`numeros=[]`
	`numeros.extend(impares)`
	`numeros.extend(pares)`
	`numeros.sort()`
		X	**C**

D	`impares=[15,7,11,3]`
	`pares=[22,14]`
	`pares.extend(impares)`
	`numeros=impares`
	`numeros.reverse()`
			D

Ejercicio 5 – Tema 6 | **1 Pto**

¿Qué devuelve la función **fun** en este programa?

```
def fun(t):
    vocales='AEIOUÁÉÍÓÚÜ'
    repes=False
    repetidas=''
    for letra in t:
        if  letra in vocales and t.count(letra)>0:
            repes=True
            repetidas+=letra
    return repetidas

txt='Estaba LA RANA cantando debajo del AGUA'
print(fun(txt))
```

Nota:
El método ***count()*** cuenta cuántas apariciones de su argumento hay en una string
Ej:
'hola, qué tal?'.count('a') -> 2

A	`'ESTABA LA RANA CANTANDO DEBAJO DEL AGUA'`		**A**
B	`False`		**B**
C	`True`		**C**
D	`'EAAAAUA'`	**X**	**D**

Ejercicio 6 – Tema 6 | **2 Ptos**

¿Qué muestra por pantalla el siguiente programa?

```
digitos='123456'
izq=' '*(len(digitos)-1)
for i in range(len(digitos)-1,-1,-1):
  print(izq+digitos[i:])
  izq=izq[1:]
```

Notas:
Se puede seleccionar una porción de una string con **[desde:hasta]**, (recordá que **hasta** no entra)
Ej:
'hola, qué tal?'[4:] -> ', qué tal?'

El operador * aplicado a una string es repetición
Ej:
'hola'*3 -> 'holaholahola'

A	`     6`
	`    56`
	`   456`
	`  3456`
	` 23456`
	`123456`
		X	**A**

B	`123456`
	` 23456`
	`  3456`
	`   456`
	`    56`
	`     6`
			B

C	`123456` `123456` `123456` `123456` `123456` `123456`		**C**
D	`16` `16` `16` `16` `16` `16`		**D**

Ejercicio 7 – Tema 6 **2 Ptos**

¿Cuál programa muestra el siguiente texto como salida?

1, 3 y 5 NO SON PARES. En cambio el resto, menos 0, si.

Notas:
Se puede seleccionar un caracter de una string de atrás para adelante con un índice negativo
Ej:
'hola, qué tal?'[-1] -> '?'

Se puede seleccionar invertir una string con **[::-1]**
Ej:
'hola?'[::-1] -> '?aloh'

A

```
numeros='012345'
pares=' son pares.'
niega='no'
frase=numeros[2]+' y '+numeros[4]
frase=frase+pares+' En cambio el resto '
frase=frase+' '+niega.upper()+pares
print(frase)
```

A

B

```
numeros='012345'
pares=' son pares.'
niega='no'
frase=numeros[-2]+' y '+numeros[-4]
frase=''
frase=frase+pares+' En cambio el resto '
frase=' '+niega.lower()+pares
print(frase)
```

B

C

```
numeros='012345'
numeros=numeros[::-1]
pares=' son pares.'
niega='no'
frase=numeros[1]+', '+numeros[3]+' y '+numeros[5]
frase=frase+' '+niega+pares
frase=frase+' En cambio el resto, menos 0, si.'
print(frase)
```

C

D

```
numeros='012345'
pares=' son pares.'
niega='no'
frase=numeros[1]+', '+numeros[3]+' y '+numeros[5]
frase=frase+' '+niega.upper()+pares.upper()
frase=frase+' En cambio el resto, menos 0, si.'
print(frase)
```

X

D

Ejercicio 8 – Tema 6 **2 Ptos**

¿Cuál es la versión correcta de la función **edita**? El programa debe mostrar
media docena es 6 , el par son 2

```
def edita(…):
  -
  -
  -

txt='Media docena es seis , el par son dos'
letra=['cero','uno','dos','tres','cuatro','cinco','seis']
print(edita(txt,letra))
```

Notas:
El método ***split()*** genera una lista con las partes de una string separadas por blancos
Ej:
txt='hola qué tal a todos'
lis= txt.split() lis-> ['hola', 'qué', 'tal', 'a', 'todos']

El método ***join()*** genera una string con los caracteres que tiene como elementos una lista separándolos por el texto al inicio
Ej:
lis= ['h', 'o', 'l', 'a']
txt=''.join(lis)-> txt='h**o**l**a'**

El método ***index()*** devuelve la posición de la primera ocurrencia del argumento en la lista
Ej:
lis= ['h', 'o', 'l', 'a', 'o']
lis.index('o')-> 1

lis= [2, 1, 8, 5]
lis.index(8)-> 2

A

```
def edita():
    t=t.upper()
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in letra:
            pos=letra.index(lista[i])
            lista[i]=str(pos)
    ' '.join(lista)
    return t
```

A

B

```
def edita(t,letra):
    lista=t.split()
    for i in range(len(lista)):
        if lista[i].lower() in letra:
            donde=letra.index(lista[i])
            lista[i]=str(donde)
    return ' '.join(lista)
```

X

B

C

```
def edita(t,letra):
    lista=t.split()
    for pal in lista:
        if pal in letra:
            pos=letra.index(pal)
            pal=str(pos)
    t=' '.join(lista)
    return letra
```

C

D

```
def edita(letra,t):
    t.lower()
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in letra:
            pos=letra.index(lista[i])
            lista[i]=str(pos)
    t=' '.join(lista)
```

D

Ejercicio 9 – Tema 6	**2 Ptos**

¿Cuál condición es candidata a ser equivalente a la siguiente?

(num in nega or num in posi) and (num%5==0 or num%4==0)

Atención!!
Te recomendamos que pruebes las condiciones con los siguientes valores:

```
nega=[-2,-21,-1]
posi=[33,5]
num=5 y num=-20
```

Nota:
Una condición (proposición) **p** es equivalente a otra **q**, si sus tablas de verdad coinciden. En otras palabras: si siempre que **p** da **V**, **q** da **V** y siempre que **p** da **F**, **q** da **F**

A	`(num not in nega and num not in posi and (num%5==0 or num%4==0))`		**A**
B	`((num%5==0 or num%4==0) and num in nega and num in posi)`		**B**
C	`((num%5<1 or num%4<1) and (num in posi or num in nega))`	**X**	**C**
D	`(num%5>=0 and num%4>=0 and (num not in posi or num in posi))`		**D**

PENSAMIENTO COMPUTACIONAL (90) .UBAXXI

EXAMEN: PRIMER PARIAL – PRIMER CUATRIMESTRE 2025 **TEMA 5**

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	NOTA Y FIRMA DOCENTE (no rellenar)
TEL:	
AULA:	

Duración del examen: 1:30h.

- ✔ Escribir claramente el nombre en todas las páginas.
- ✔ El examen consta de 9 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 5 inclusive permiten acumular 1 punto (si son correctas), de la 6 a la 9 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**

Consignar con una X (cruz) la respuesta elegida en cada ejercicio en la siguiente matriz:

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Talón de Control para el Alumno (Tema 5)

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Ejercicio 1 – Tema 5			**1 Pto**
¿ Cuál de los siguientes programas muestra por pantalla **solo importado**?			

A	`producto='heladera'`
	`proveedor='Lider s.a.'`
	`precio=750000`
	`if (producto =='freezer vertical' and`
	`  producto=='heladera' and`
	`  producto=='lavavajillas'):`
	`  if proveedor in ('lastrech','dami','electrtech'):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`print('solo nacional')`
			A

B	`producto='heladera'`
	`proveedor='Lider s.a.'`
	`precio=750000`
	`if (producto =='heladera no frost' or`
	`  producto=='ventilador' or`
	`  producto=='lavarropas'):`
	`  if (proveedor in ('lastrech','dami',`
	`  'electrtech')):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`elif (producto in ('freezer horizontal',`
	`  'freezer vertical',`
	`  'anafe')):`
	`  print('solo nacional')`
			B

C	`producto='heladera'`
	`proveedor='Lider s.a.'`
	`precio=750000`
	`if producto in ('heladera','freezer vertical'):`
	`  if (proveedor not in ('lastrech','dami',`
	`  'atlas')):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`elif (producto in ('anafe','lavavajillas',`
	`  'lavaropas')):`
	`  print('solo nacional')`
		X	**C**

D	`producto='heladera'`
	`proveedor='Lider s.a.'`
	`precio=750000`
	`if producto in ('horno','anafe'):`
	`  if proveedor!='Lider s.a.':`
	`  print('solo importado')`
	`  else:`
	`  print('nacional e importado')`
	`else:`
	`  print('solo nacional')`
			D

Ejercicio 2 – Tema 5			1 Pto

¿Qué muestra el siguiente programa?
 	
`a=0.5`
`b=-1`
`c=1.5`
`d=2`
`print((a+d*b)/c-b)`			

A	`0.4`		**A**
B	`2.5`		**B**
C	`0.0`	**X**	**C**
D	`-2.0`		**D**

Ejercicio 3 – Tema 5			1 Pto

¿Con qué contenido acabará la lista **final** en el siguiente programa?
 	
`indicador=[0,6,5,0,6,5]`
`numeros=[1,2,3,4,5,6]`
`final=[0]*10`
`for i in range(len(numeros)):`
`  final[indicador[i]]=numeros[i]`
 	
Nota:
El operador * aplicado a una lista es repetición
Ej:
[1, 3]*2 -> [1, 3, 1, 3]			

A	`[0, 0, 0, 0, 0, 0]`		**A**
B	`[1, 0, 0, 0, 0, 3, 2, 0, 5, 6]`		**B**
C	`[4, 0, 0, 0, 0, 6, 5, 0, 0, 0]`	**X**	**C**
D	`[6, 5, 4, 3, 2, 1]`		**D**

Ejercicio 4 – Tema 5 | **1 Pto**

¿Cuál programa integra **ordenadamente** en la lista **numeros** tanto los pares como los impares? En **numeros** debe quedar **[3, 7, 11, 14, 15, 22]**

Notas:
El método ***reverse()*** invierte una lista
Ej:
lis=[-1,2,0]
lis.reverse() lis-> [0,2,-1]

El método ***sort()*** ordena una lista
Ej:
lis=[6,-1,0]
lis.sort() -> lis=[-1, 0, 6]

El método ***extend()*** agrega una lista a otra
Ej:
lis1=[6,-1,5,6]
lis2=[0,0]
lis1.extend(lis2) -> lis1=[6, -1, 5, 6, 0, 0]

El método ***insert(posición,nuevo elemento)*** permite agregar un elemento en una lista en cualquier posición
Ej:
lis=[1, 2, 3]
lis.insert(1,8) -> lis=[1, 8, 2, 3]

A

```
impares=[15,7,11,3]
pares=[22,14]
numeros=[]
for i in range(len(pares)):
    numeros.append(impares[i])
    numeros.append(pares[i])
```

A

B

```
impares=[15,7,11,3]
pares=[22,14]
numeros=impares
for num in pares:
    if num not in numeros:
        numeros.insert(0,num)
```

B

C

```
impares=[15,7,11,3]
pares=[22,14]
pares.extend(impares)
numeros=pares
numeros.reverse()
```

C

D

```
impares=[15,7,11,3]
pares=[22,14]
numeros=impares+pares
numeros.sort()
```

X **D**

Ejercicio 5 – Tema 5 **1 Pto**

¿Qué devuelve la función **fun** en este programa?

```
def fun(t):
    vocales='AEIOUÁÉÍÓÚÜ'
    repes=False
    repetidas=''
    for letra in t:
        if  letra in vocales and t.count(letra)>0:
            repes=True
            repetidas+=letra
    return repes

txt='Estaba LA RANA cantando debajo del AGUA'
print(fun(txt))
```

Nota:
El método ***count()*** cuenta cuántas apariciones de su argumento hay en una string
Ej:
'hola, qué tal?'.count('a') -> 2

A	`'ESTABA LA RANA CANTANDO DEBAJO DEL AGUA'`		**A**
B	`False`		**B**
C	`True`	**X**	**C**
D	`'EAAAAUA'`		**D**

Ejercicio 6 – Tema 5 **2 Ptos**

¿Qué muestra por pantalla el siguiente programa?

```
digitos='123456'
izq=''
for i in range(len(digitos)):
  print(izq+digitos)
  digitos=digitos[1:]
  izq+=' '
```

Notas:
Se puede seleccionar una porción de una string con **[desde:hasta]**, (recordá que **hasta** no entra)
Ej:
'hola, qué tal?'[4:] -> ', qué tal?'

A	`     6`
	`    56`
	`   456`
	`  3456`
	` 23456`
	`123456`
			A

B	`123456`
	` 23456`
	`  3456`
	`   456`
	`    56`
	`     6`
		X	**B**

C	`123456` `123456` `123456` `123456` `123456` `123456`		**C**
D	`16` `16` `16` `16` `16` `16`		**D**

Ejercicio 7 – Tema 5 **2 Ptos**

¿Cuál programa muestra el siguiente texto como salida?

2 y 4 son pares. En cambio el resto NO son pares.

Notas:
Se puede seleccionar un caracter de una string de atrás para adelante con un índice negativo
Ej:
'hola, qué tal?'[-1] -> '?'

Se puede seleccionar invertir una string con **[::-1]**
Ej:
'hola?'[::-1] -> '?aloh'

A

```
numeros='012345'
pares=' son pares.'
niega='no'
frase=numeros[2]+' y '+numeros[4]
frase=frase+pares+' En cambio el resto '
frase=frase+' '+niega.upper()+pares
print(frase)
```

X **A**

B

```
numeros='012345'
pares=' son pares.'
niega='no'
frase=numeros[-2]+' y '+numeros[-4]
frase=''
frase=frase+pares+' En cambio el resto '
frase=' '+niega.lower()+pares
print(frase)
```

B

C

```
numeros='012345'
numeros=numeros[::-1]
pares=' son pares.'
niega='no'
frase=numeros[1]+', '+numeros[3]+' y '+numeros[5]
frase=frase+' '+niega+pares
frase=frase+' En cambio el resto, menos 0, si.'
print(frase)
```

C

D

```
numeros='012345'
pares=' son pares.'
niega='no'
frase=numeros[1]+', '+numeros[3]+' y '+numeros[5]
frase=frase+' '+niega.upper()+pares.upper()
frase=frase+' En cambio el resto, menos 0, si.'
print(frase)
```

D

Ejercicio 8 – Tema 5 **2 Ptos**

¿Cuál es la versión correcta de la función **edita**? El programa debe mostrar
si no es 1 es 2 , pero siempre más que 0

```
def edita(…):
  -
  -
  -

txt='Si no es UNo es dos , pero siempre más que CERo'
letra=['cero','uno','dos','tres','cuatro']
print(edita(txt,letra))
```

Notas:
El método ***split()*** genera una lista con las partes de una string separadas por blancos
Ej:
txt='hola qué tal a todos'
lis= txt.split() lis-> ['hola', 'qué', 'tal', 'a', 'todos']

El método ***join()*** genera una string con los caracteres que tiene como elementos una lista separándolos por el texto al inicio
Ej:
lis= ['h', 'o', 'l', 'a']
txt=''.join(lis)-> txt='h**o**l**a'**

El método ***index()*** devuelve la posición de la primera ocurrencia del argumento en la lista
Ejs:
lis= ['h', 'o', 'l', 'a', 'o'] **lis= [2, 1, 8, 5]**
lis.index('o')-> 1 **lis.index(8)-> 2**

A

```
def edita():
    t=t.upper()
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in letra:
            pos=letra.index(lista[i])
            lista[i]=str(pos)
    ' '.join(lista)
    return t
```

A

B

```
def edita(t,letra):
    t=t.lower()
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in letra:
            pos=letra.index(lista[i])
            lista[i]=str(pos)
    t=' '.join(lista)
    return t
```

X

B

C

```
def edita(t,letra):
    lista=t.split()
    for pal in lista:
        if pal in letra:
            pos=letra.index(pal)
            pal=str(pos)
    t=' '.join(lista)
    return letra
```

C

D

```
def edita(letra,t):
    t.lower()
    lista=t.split()
    for i in range(len(lista)):
        if lista[i] in letra:
            pos=letra.index(lista[i])
            lista[i]=str(pos)
    t=' '.join(lista)
```

D

Ejercicio 9 – Tema 5	**2 Ptos**

¿Cuál condición es candidata a ser equivalente a la siguiente?

(num in nega or num in posi) and (num%5==0 or num%4==0)

Atención!!
Te recomendamos que pruebes las condiciones con los siguientes valores:

```
nega=[-2,-21,-1]
posi=[33,5]
num=5 y num=-20
```

Nota:
Una condición (proposición) **p** es equivalente a otra **q**, si sus tablas de verdad coinciden. En otras palabras: si siempre que **p** da **V**, **q** da **V** y siempre que **p** da **F**, **q** da **F**

A	`((num%5==0 or num%4==0) and num in nega and num in posi)`		**A**
B	`(num not in nega and num not in posi and (num%5==0 or num%4==0))`		**B**
C	`(num%5>=0 and num%4>=0 and (num not in posi or num in posi))`		**C**
D	`(((num%5==0 or num%4==0) and num in nega) or (num in posi and (num%5==0 or num%4==0)))`	**X**	**D**

PENSAMIENTO COMPUTACIONAL (90) .UBAXXI

EXAMEN: PRIMER PARIAL – PRIMER CUATRIMESTRE 2025 **TEMA 4**

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	NOTA Y FIRMA DOCENTE (no rellenar)
TEL:	
AULA:	

Duración del examen: 1:30h.

- ✔ Escribir claramente el nombre en todas las páginas.
- ✔ El examen consta de 9 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 5 inclusive permiten acumular 1 punto (si son correctas), de la 6 a la 9 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**

Consignar con una X (cruz) la respuesta elegida en cada ejercicio en la siguiente matriz:

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Talón de Control para el Alumno (Tema 4)

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Ejercicio 1 – Tema 4			**1 Pto**
¿ Cuál de los siguientes programas muestra por pantalla **solo importado**?			

A	`producto='decodificador'`
	`proveedor='Lider s.a.'`
	`precio=275000`
	`if (producto =='plaqueta m33' and`
	`producto=='decodificador' and`
	`producto=='antena'):`
	`  if proveedor in ('lastrech','dami','electrtech'):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`print('solo nacional')`
			A

B	`producto='decodificador'`
	`proveedor='Lider s.a.'`
	`precio=275000`
	`if (producto =='plaqueta m33' or`
	`producto=='decodificador' or`
	`producto=='antena'):`
	`  if (proveedor in ('lastrech','dami',`
	`'electrtech')):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`elif (producto in ('plaqueta m22','switcher',`
	`'cable 11d')):`
	`  print('solo nacional')`
			B

C	`producto='decodificador'`
	`proveedor='Lider s.a.'`
	`precio=275000`
	`if producto in ('decodificador','antena'):`
	`  if proveedor not in ('lastrech','dami',`
	`'electrtech'):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`elif producto in ('plaqueta m22','switcher',`
	`'cable 11d'):`
	`  print('solo nacional')`
		X	**C**

D	`producto='decodificador'`
	`proveedor='Lider s.a.'`
	`precio=275000`
	`if producto in ('antena wifi','antena'):`
	`  if proveedor!='Lider s.a.':`
	`  print('solo importado')`
	`  else:`
	`  print('nacional e importado')`
	`else:`
	`  print('solo nacional')`
			D

Ejercicio 2 – Tema 4			1 Pto

¿Qué muestra el siguiente programa?

```
a=0
b=1
c=2
d=-2
cant=d*d+b**(2+c)*a
print('0'*cant)
```

Nota:
El operador ** es potencia
Ej:
13 -> 1** y **5**2 -> 25**

El operador * aplicado a una string y un número entero es repetición
Ej:
'hola'*2-> 'holahola'

A	000		**A**
B	0		**B**
C	00000000		**C**
D	0000	**X**	**D**

Ejercicio 3 – Tema 4			1 Pto

¿Con qué contenido acabará la lista **final** en el siguiente programa?

```
indicador=[7,5,1,3,2,4,6]
numeros=[1,2,3,4,5,6,7]
final=[0]*7
for i in range(len(numeros)):
    final[indicador[i]-1]=numeros[i]
```

Nota:
El operador * aplicado a una lista es repetición
Ej:
[1, 3]*2 -> [1, 3, 1, 3]

A	[0, 1, 2, 3, 4, 5, 6, 7]		**A**
B	[3, 5, 4, 6, 2, 7, 1]	**X**	**B**
C	[0, 0, 0, 0, 0, 0, 0, 0]		**C**
D	[4, 5, 7, 6, 3, 2, 1]		**D**

Ejercicio 4 – Tema 4	**1 Pto**

¿Cuál programa guarda en la lista **naturales** únicamente los números naturales y los almacena en **orden invertido** al original? En naturales debe quedar **[33, 22, 15]**
Recordá que un número es natural si es entero mayor que cero

Notas:
El método ***reverse()*** invierte una lista
Ej:
lis=[-1,2,0]
lis.reverse() lis-> [0,2,-1]

El método ***copy()*** genera una copia de la lista
Ej:
lis=[6,0]
lis1=lis.copy() -> hay dos listas que en ese momento tienen los mismos datos

El método ***remove()*** quita la primer ocurrencia de un elemento en una lista
Ej:
lis=[6,-1,0,5,6]
lis.remove(6) -> lis=[-1, 0, 5, 6]

El método ***insert(posición,nuevo elemento)*** permite agregar un elemento en una lista en cualquier posición
Ej:
lis=[1, 2, 3]
lis.insert(1,8) -> lis=[1, 8, 2, 3]

A	`numeros=[15,-2,20.5,22,0,33,-1]`
	`naturales=[]`
	`for num in numeros:`
	`  if int(num)!=num:`
	`  naturales.append(num)`
			A

B	`numeros=[15,-2,20.5,22,0,33,-1]`
	`naturales=[]`
	`for num in numeros:`
	`  if num>0:`
	`  naturales.append(num)`
	`numeros.reverse()`
			B

C	`numeros=[15,-2,20.5,22,0,33,-1]`
	`naturales=[]`
	`for num in numeros:`
	`  if int(num)==num and num>0:`
	`  naturales.insert(0,num)`
		X	**C**

D	`numeros=[15,-2,20.5,22,0,33,-1]`
	`naturales=numeros.copy()`
	`for num in numeros:`
	`  if num>0 and int(num)==num:`
	`  naturales.remove(num)`
			D

Ejercicio 5 – Tema 4 — **1 Pto**

¿Qué devuelve la función **fun** en este programa?

```
def fun(t):
    vocales='AEIOUÁÉÍÓÚÜ'
    t=t.upper()
    cont=0
    for letra in t:
        if letra in vocales:
            cont+=1
    if cont>6:
        resp=False
    else:
        resp=True
    return resp

txt='Una oruga NEGRA en un PAJAR'
print(fun(txt))
```

A	`(11)`		**A**
B	`('UNA ORUGA NEGRA EN UN PAJAR', 11)`		**B**
C	`False`	**X**	**C**
D	`('Una oruga NEGRA en un PAJAR',)`		**D**

Ejercicio 6 – Tema 4 — **2 Ptos**

¿Qué muestra por pantalla el siguiente programa?

```
cartel='desde'
print(cartel.upper())
izq=' '
for i in range(1,len(cartel)-1):
  print(izq+cartel[i])
  izq+=' '
print(cartel.upper())
```

A	`desde` `edsed`		**A**

B	`DESDE`
	`D  E`
	`D  E`
	`D  E`
	`DESDE`
			B

C	`DESDE`
	`  e`
	`  s`
	`  d`
	`DESDE`
		X	**C**

D	`DESDEdesde`		**D**

Ejercicio 7 – Tema 4 **2 Ptos**

¿Cuál programa muestra el siguiente texto como salida?

Siempre la vocal: A está varias veces en ANACONDA

Nota:
Se puede seleccionar un caracter de una string de atrás para adelante con un índice negativo
Ej:
'hola, qué tal?'[-1] -> '?'

A

```
palabra='anaconda'
txt='la vocal: '
frase='Siempre '
frase=txt+palabra[1]
frase=frase+' está varias veces en '
frase=palabra
print(frase)
```

A

B

```
palabra='anaconda'
txt='la vocal: '
frase='Siempre '
frase=frase+txt+palabra[-1].upper()
frase=frase+' está varias veces en '
frase=frase+palabra.upper()
print(frase)
```

X **B**

C

```
palabra='anaconda'
txt='las vocales:'
frase='Siempre '
frase=frase+txt+palabra[2].upper()
frase=frase+' y '+palabra[-4].upper()
frase=frase+' están en '+palabra
print(frase)
```

C

D

```
palabra='anaconda'
txt='las vocales:'
frase='Siempre '
frase=frase+txt+palabra[-3]
frase=frase+' y '+palabra[2].upper()
frase=' están en '+palabra
print(frase)
```

D

Ejercicio 8 – Tema 4	2 Ptos

¿Cuál es la versión correcta de la función **edita**? El programa debe mostrar
HASTA<ESP>LA<ESP>VISTA<ESP>BABY<ESP><ESP><ESP>

```
def edita(…):
  -
  -
  -

txt='hasta la vista baby!!!'
print(edita(txt,',- .!?'))
```

Notas:
La función ***list()*** convierte un texto en una lista de caracteres
Ej:
txt='hola'
lis= list(txt) lis-> ['h', 'o', 'l', 'a']

El método ***join()*** genera una string con los caracteres que tiene como elementos una lista separándolos por el texto al inicio
Ej:
lis= ['h', 'o', 'l', 'a']
txt=''.join(lis)-> txt='h**o**l**a'**

A

```
def edita(carEsp):
    lista=list(t)
    for i in range(len(lista)):
        if lista[i] in carEsp:
            lista[i]='<esp>'
    t=''.join(lista)
    return lista
```

B

```
def edita(t,carEsp):
    lista=list(t)
    for i in range(len(lista)):
        if lista[i] in carEsp:
            lista[i]='<esp>'
    t=''.join(lista)
    return t.upper()
```

X

C

```
def edita(t,carEsp):
    lista=list(t)
    for i in range(len(lista)):
        if lista[i] in carEsp:
            lista[i]=''
    t=''.join(lista)
    return t
```

D

```
def edita(carEsp,t):
    lista=list(t)
    for i in range(len(lista)):
        if lista[i] in carEsp:
            lista[i]='<esp>'
    t=''.join(lista)
```

Ejercicio 9 – Tema 4	**2 Ptos**

¿Cuál condición es candidata a ser equivalente a la siguiente?

(num in sel) and (num!=int(num) or num>0 or num%3==0)

Atención!!
Te recomendamos que pruebes las condiciones con los siguientes valores:

```
sel=[2,21,-1,56]
num=22 y num=21
```

Nota:
Una condición (proposición) **p** es equivalente a otra **q**, si sus tablas de verdad coinciden. En otras palabras: si siempre que **p** da **V**, **q** da **V** y siempre que **p** da **F**, **q** da **F**

A	`((num>=1 or not(int(num)==num) or (num%3!=1 and num%3!=2)) and (num in sel))`	X	A
B	`num in sel or num!=int(num) or num>0 or num%3==0`		B
C	`num!=int(num) or num>0 or num%3==0`		C
D	`(num in sel and num!=int(num) and num>0 and num%3==0)`		D

PENSAMIENTO COMPUTACIONAL (90) .UBAXXI

EXAMEN: PRIMER PARIAL – PRIMER CUATRIMESTRE 2025 **TEMA 2**

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	NOTA Y FIRMA DOCENTE (no rellenar)
TEL:	
AULA:	

Duración del examen: 1:30h.

- ✔ Escribir claramente el nombre en todas las páginas.
- ✔ El examen consta de 9 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 5 inclusive permiten acumular 1 punto (si son correctas), de la 6 a la 9 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**

Consignar con una X (cruz) la respuesta elegida en cada ejercicio en la siguiente matriz:

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Talón de Control para el Alumno (Tema 2)

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Ejercicio 1 – Tema 2			**1 Pto**
¿Cuál de los siguientes programas muestra por pantalla **caro**?			

A	`producto='clavo'`
	`unidad='caja'`
	`precio=275`
	`if producto in ('clavo','manguera','tornillo'):`
	`  if precio<1000 and unidad=='caja' and unidad=='kg':`
	`  print('carito')`
	`  elif precio<1000 and unidad=='unidad':`
	`  print('caro')`
	`elif producto in ('cola','pincel','lija'):`
	`  print('barato')`
			A

B	`producto='clavo'`
	`unidad='caja'`
	`precio=275`
	`if producto in ('manguera','tornillo'):`
	`  if precio>1000 and (unidad=='caja' or`
	`  unidad=='kg'):`
	`  print('carito')`
	`  elif precio<1000 and unidad=='unidad':`
	`  print('maso')`
	`elif producto in ('cola','lija','clavo'):`
	`  if unidad in ('ciento','docena','rollo'):`
	`  print('barato')`
	`  else:`
	`  print('caro')`
		X	**B**

C	`producto='clavo'`
	`unidad='caja'`
	`precio=275`
	`if producto in ('clavo','manguera','tornillo'):`
	`  if precio>1000 and`
	`  (unidad=='caja' or unidad=='kg'):`
	`  print('carito')`
	`  elif precio>1000 and unidad=='unidad':`
	`  print('caro')`
	`print('barato')`
			C

D	`producto='clavo'`
	`unidad='caja'`
	`precio=275`
	`if producto in ('clavo','manguera','tornillo'):`
	`  if precio>1000 or unidad=='caja'`
	`  or unidad=='kg':`
	`  print('carito')`
	`  elif precio>1000 and unidad=='unidad':`
	`  print('caro')`
	`else:`
	`  print('barato')`
			D

Ejercicio 2 – Tema 2 **1 Pto**

¿Qué muestra el siguiente programa?

```
a=0
b=1
c=1.5
d=-2
print((a+d**2)/(c-b))
```

Nota:
El operador ** es potencia
Ej:
13 -> 1** y **5**2 -> 25**

A	0.4		**A**
B	8.0	**X**	**B**
C	2.0		**C**
D	-5.0		**D**

Ejercicio 3 – Tema 2 **1 Pto**

¿Qué contenido tendrán al final las listas **a**, **b** y **c** en el siguiente programa?

```
indicador=[2,2,2,3,3,2,2]
numeros=[1,2,3,4,5,6,7]
a=[]
b=[]
c=[]
for i in range(len(numeros)):
    if indicador[i]==1:
        a.append(numeros[i])
    elif indicador[i]==2:
        b.append(numeros[i])
    else:
        c.append(numeros[i])
```

A	a [] b [1, 2, 3, 6, 7] c [4, 5]	**X**	**A**
B	a [2, 7] b [1, 3] c [4, 5, 6]		**B**
C	a [0, 0] b [2, 2] c [3, 3, 3]		**C**
D	a [1, 7] b [4, 5, 6] c [2, 3]		**D**

Ejercicio 4 – Tema 2	**1 Pto**

¿Cuál programa guarda en la lista **pares** únicamente los números pares y los almacena en **orden invertido** al original? En pares debe quedar **[22, 20]**
Recordá que un número es par si es mayor que cero y divisible por 2

Notas:
El método ***reverse()*** invierte una lista
Ej:
lis=[-1,2,0]
lis.reverse() lis-> [0,2,-1]

El método ***sort()*** ordena una lista
Ej:
lis=[6,-1,0]
lis.sort() -> lis=[-1, 0, 6]

El método ***insert(posición,nuevo elemento)*** permite agregar un elemento en una lista en cualquier posición
Ej:
lis=[1, 2, 3]
lis.insert(1,8) -> lis=[1, 8, 2, 3]

A

```
numeros=[15,-2,20,22,0,33,-1]
pares=[]
for num in numeros:
    if num>0 and num%2!=0:
        pares.append(num)
```

A

B

```
numeros=[15,-2,20,22,0,33,-1]
pares=[]
for num in numeros:
    if num<0 and num%2==0:
        pares.append(num)
numeros.reverse()
```

B

C

```
numeros=[15,-2,20,22,0,33,-1]
pares=[]
for num in numeros:
    if num%2!=1 and num>=1:
        pares.insert(0,num)
```

X

C

D

```
numeros=[15,-2,20,22,0,33,-1]
pares=[]
for num in numeros:
    if num>0:
        pares.insert(0,num)
pares.sort()
```

D

Ejercicio 5 – Tema 2 — **1 Pto**

¿Qué devuelve la función **fun** en este programa?

```
def fun(t):
    vocales='AEIOUÁÉÍÓÚÜ'
    t=t.upper()
    cont=0
    for letra in t:
        if letra in vocales:
            cont+=1
    return cont

txt='Una oruga NEGRA en un PAJAR'
print(fun(txt))
```

A	`11`	**X**	**A**
B	`0.0`		**B**
C	`True`		**C**
D	`False`		**D**

Ejercicio 6 – Tema 2 — **2 Ptos**

¿Qué muestra por pantalla el siguiente programa?

```
cartel='BARCO'
extremo=' '*(len(cartel)//2)
centro=' '
print(extremo+cartel[len(cartel)//2])
extremo=extremo[1:]
for i in range(len(cartel)//2-1,-1,-1):
  j=len(extremo)+1
  print(extremo+cartel[i]+centro+cartel[-j])
  extremo=extremo[1:]
  centro+='  '
```

Atención!
En
`centro+='  '`
Estoy agregando **2** espacios a centro ¡!!!

Notas:
Se puede seleccionar una porción de una string con **[desde:hasta]**, (recordá que **hasta** no entra)
Ej:
'hola, qué tal?'[4:] -> ', qué tal?'

El operador // es el cociente entero
Ej:
7//2 -> 3

El operador * aplicado a una string es repetición
Ej:
'hola'*3 -> 'holaholahola'

A	`b   o`
	` a c`
	`  r`
			A

B	`BARCO` `BARCO`		**B**

C	`B`
	` A`
	`  R`
	`   C`
	`    O`
			C

D	`  R`
	` A C`
	`B   O`
		X	**D**

Ejercicio 7 – Tema 2			2 Ptos
¿Cuál programa muestra el siguiente texto como salida? **LA PLACA bacteriana DENTAL SE QUITA CON PLAKOUT, AGUSTÍN** **Nota:** Se puede seleccionar una porción de una string con **[desde:hasta]**, (recordá que **hasta** no entra) **Ej:** **'hola, qué tal?'[4:] -> ', qué tal?'**			

A	`mens='LA PLACA DENTAL SE QUITA CON'`
	`nom='Agustín'`
	`tipo='bacteriana'`
	`producto='plakout'`
	`textoMens=mens[:8]+' '+tipo`
	`textoMens=textoMens+mens[8:]`
	`textoMens=textoMens+' '+producto.upper()`
	`textoMens=textoMens+', '+nom.upper()`
	`print(textoMens)`
		X	**A**

B	`mens='LA PLACA DENTAL SE QUITA CON'`
	`nom='Agustín'`
	`tipo='bacteriana'`
	`producto='plakout'`
	`textoMens=nom+', '+mens[:8]+' '+tipo.upper()`
	`textoMens=textoMens+mens[8:]`
	`textoMens=textoMens+': '+producto.upper()`
	`print(textoMens)`
			B

C	`mens='LA PLACA DENTAL SE QUITA CON'`
	`nom='Agustín'`
	`tipo='bacteriana'`
	`producto='plakout'`
	`textoMens=mens[:8]+' '+tipo`
	`textoMens=mens[8:]`
	`textoMens=' '+producto`
	`textoMens=textoMens+', '+nom`
	`print(textoMens)`
			C

D	`mens='LA PLACA DENTAL SE QUITA CON'`
	`nom='Agustín'`
	`tipo='bacteriana'`
	`producto='plakout'`
	`textoMens=nom.upper()+', '+mens[8]+' '+tipo`
	`textoMens=mens[8]`
	`textoMens=textoMens+': '+producto.upper()`
	`print(textoMens)`
			D

Ejercicio 8 – Tema 2 | **2 Ptos**

¿Cuál es la versión correcta de la función **sinLetras**? El programa debe mostrar
ST HSTR CNTNRÁ...

```
def sinLetras(…):
  -
  -
  -

txt='esta historia continuará...'
print(sinLetras(txt,'aeiou'))
```

Notas:
La función ***list()*** convierte un texto en una lista de caracteres
Ej:
txt='hola'
lis= list(txt) lis-> ['h', 'o', 'l', 'a']

El método ***join()*** genera una string con los caracteres que tiene como elementos una lista separándolos por el texto al inicio
Ej:
lis= ['h', 'o', 'l', 'a']
txt=''.join(lis)-> txt='h**o**l**a'**

El método ***remove()*** quita la primer ocurrencia de un elemento en una lista
Ej:
lis=[6,-1,0,5,6]
lis.remove(6) -> lis=[-1, 0, 5, 6]

A

```
def sinLetras(t,let):
    lista=list(t)
    for car in let:
       lista.remove(car)
    t=''.join(lista)
    return t
```

B

```
def sinLetras():
    lista=list(t)
    for car in let:
        while car in lista:
            lista.remove(car)
    return lista
```

C **X**

```
def sinLetras(t,let):
    lista=list(t)
    for car in let:
        while car in lista:
            lista.remove(car)
    t=''.join(lista)
    return t.upper()
```

D

```
def sinLetras(t):
    lista=list(t)
    for car in let:
        while car in lista:
            lista.remove(car)
    ''.join(lista)
```

Ejercicio 9 – Tema 2	**2 Ptos**

¿Cuál condición es candidata a ser equivalente a la siguiente?

(num>0 and num==int(num) and num%3==0) or (num in sel)

Atención!!
Te recomendamos que pruebes las condiciones con los siguientes valores:

```
sel=[2,25,-1,56]
num=22 y num=21
```

Nota:
Una condición (proposición) **p** es equivalente a otra **q**, si sus tablas de verdad coinciden. En otras palabras: si siempre que **p** da **V**, **q** da **V** y siempre que **p** da **F**, **q** da **F**

A	`((num in sel) or (int(num)==num and num>=1 and num%3 in [0]))`	X	A
B	`num not in sel or num==int(num)`		B
C	`(num in sel and num==int(num) and num>0 and num%3<1)`		C
D	`num in sel or num==int(num) or num>0 or num%3<1`		D

PENSAMIENTO COMPUTACIONAL (90) .UBAXXI

EXAMEN: PRIMER PARIAL – PRIMER CUATRIMESTRE 2025 **TEMA 3**

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	NOTA Y FIRMA DOCENTE (no rellenar)
TEL:	
AULA:	

Duración del examen: 1:30h.

- ✔ Escribir claramente el nombre en todas las páginas.
- ✔ El examen consta de 9 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 5 inclusive permiten acumular 1 punto (si son correctas), de la 6 a la 9 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**

Consignar con una X (cruz) la respuesta elegida en cada ejercicio en la siguiente matriz:

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Talón de Control para el Alumno (Tema 3)

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Ejercicio 1 – Tema 3			**1 Pto**
¿ Cuál de los siguientes programas muestra por pantalla **nacional e importado**?			

A	`producto='decodificador'`
	`proveedor='Lider s.a.'`
	`precio=275000`
	`if producto =='plaqueta m33' and`
	`  producto=='decodificador' and producto=='antena':`
	`  if proveedor in ('lastrech','dami','electrtech'):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`print('solo nacional')`
			A

B	`producto='decodificador'`
	`proveedor='Lider s.a.'`
	`precio=275000`
	`if producto =='plaqueta m33' or`
	`  producto=='decodificador' or`
	`  producto=='antena':`
	`  if proveedor in ('lastrech','dami',`
	`  'electrtech'):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`elif producto in ('plaqueta m22','switcher',`
	`  'cable 11d'):`
	`  print('solo nacional')`
		X	**B**

C	`producto='decodificador'`
	`proveedor='Lider s.a.'`
	`precio=275000`
	`if producto in ('decodificador','antena'):`
	`  if proveedor not in ('lastrech','dami',`
	`  'electrtech'):`
	`  print('solo importado')`
	`  elif precio<1000000:`
	`  print('nacional e importado')`
	`elif producto in ('plaqueta m22','switcher',`
	`  'cable 11d'):`
	`  print('solo nacional')`
			C

D	`producto='decodificador'`
	`proveedor='Lider s.a.'`
	`precio=275000`
	`if producto in ('antena wifi','antena'):`
	`  if proveedor!='Lider s.a.':`
	`  print('solo importado')`
	`  else:`
	`  print('nacional e importado')`
	`else:`
	`  print('solo nacional')`
			D

Ejercicio 2 – Tema 3	1 Pto

¿Qué muestra el siguiente programa?

```
a=0
b=1
c=2
d=-2
cant=a*d+b**2+c*b
print('0'*cant)
```

Nota:
El operador ** es potencia
Ej:
13 -> 1** y **5**2 -> 25**

El operador * aplicado a una string y un número entero es repetición
Ej:
'hola'*2-> 'holahola'

A	000	**X**	**A**
B	0		**B**
C	00		**C**
D	000000		**D**

Ejercicio 3 – Tema 3	1 Pto

¿Con qué contenido acabará la lista **final** en el siguiente programa?

```
indicador=[7,6,5,1,2,4,3]
numeros=[1,2,3,4,5,6,7]
final=[0]*7
for i in range(len(numeros)):
    final[indicador[i]-1]=numeros[i]
```

Nota:
El operador * aplicado a una lista es repetición
Ej:
[1, 3]*2 -> [1, 3, 1, 3]

A	[0, 1, 2, 3, 4, 5, 6, 7]		**A**
B	[3, 5, 4, 6, 2, 7, 1]		**B**
C	[0, 0, 0, 0, 0, 0, 0, 0]		**C**
D	[4, 5, 7, 6, 3, 2, 1]	**X**	**D**

Ejercicio 4 – Tema 3	**1 Pto**

¿Cuál programa guarda en la lista **naturales** únicamente los números naturales y los almacena en **orden invertido** al original? En naturales debe quedar **[33, 22, 15]**
Recordá que un número es natural si es entero mayor que cero

Notas:
El método ***reverse()*** invierte una lista
Ej:
lis=[-1,2,0]
lis.reverse() lis-> [0,2,-1]

El método ***remove()*** quita la primer ocurrencia de un elemento en una lista
Ej:
lis=[6,-1,0,5,6]
lis.remove(6) -> lis=[-1, 0, 5, 6]

El método ***copy()*** genera una copia de la lista
Ej:
lis=[6,0]
lis1=lis.copy() -> hay dos listas que en ese momento tienen los mismos datos

El método ***insert(posición,nuevo elemento)*** permite agregar un elemento en una lista en cualquier posición
Ej:
lis=[1, 2, 3]
lis.insert(1,8) -> lis=[1, 8, 2, 3]

A

```
numeros=[15,-2,20.5,22,0,33,-1]
naturales=[]
for num in numeros:
    if num>0 and int(num)==num:
        naturales.append(num)
naturales.reverse()
```

X **A**

B

```
numeros=[15,-2,20.5,22,0,33,-1]
naturales=[]
for num in numeros:
    if num>0:
        naturales.append(num)
numeros.reverse()
```

B

C

```
numeros=[15,-2,20.5,22,0,33,-1]
naturales=[]
for num in numeros:
    if num<0 and int(num)!=num:
        naturales.insert(0,num)
```

C

D

```
numeros=[15,-2,20.5,22,0,33,-1]
naturales=numeros.copy()
for num in numeros:
    if num>0 and int(num)==num:
        naturales.remove(num)
```

D

Ejercicio 5 – Tema 3 | **1 Pto**

¿Qué devuelve la función **fun** en este programa?

```
def fun(t):
    vocales='AEIOUÁÉÍÓÚÜ'
    t=t.upper()
    cont=0
    for letra in t:
        if letra in vocales:
            cont+=1
    return (t,cont)

txt='Una oruga NEGRA en un PAJAR'
print(fun(txt))
```

A	`(11)`		**A**
B	`('UNA ORUGA NEGRA EN UN PAJAR', 11)`	**X**	**B**
C	`False`		**C**
D	`('Una oruga NEGRA en un PAJAR',)`		**D**

Ejercicio 6 – Tema 3 | **2 Ptos**

¿Qué muestra por pantalla el siguiente programa?

```
cartel='desde'
for i in range(len(cartel)):
  linea=''
  for j in range(len(cartel)):
    if (i%2==0 and j%2==0) or (i%2!=0 and j%2!=0):
      linea+=cartel[j].upper()
    else:
      linea+=' '
  print(linea)
```

A	`desde`
	`d   e`
	`d   e`
	`d   e`
	`desde`
			A

B	`DESDE`
	`desde`
	`DESDE`
	`desde`
	`DESDE`
			B

C	`DESDE`
	`  e`
	`   s`
	`    d`
	`DESDE`
			C

D	`D S E`
	`  E D`
	`D S E`
	`  E D`
	`D S E`
		X	**D**

Ejercicio 7 – Tema 3			**2 Ptos**
¿Cuál programa muestra el siguiente texto como salida? **Siempre las vocales:A y O están en anaconda** **Nota:** Se puede seleccionar un caracter de una string de atrás para adelante con un índice negativo **Ej:** **'hola, qué tal?'[-1] -> '?'**			

A	`palabra='anaconda'`
	`txt='la vocal: '`
	`frase='Siempre '`
	`frase=txt+palabra[1]`
	`frase=frase+' está varias veces en '`
	`frase=palabra`
	`print(frase)`
			A

B	`palabra='anaconda'`
	`txt='la vocal: '`
	`frase='Siempre '`
	`frase=frase+txt+palabra[-1].upper()`
	`frase=frase+' está varias veces en '`
	`frase=frase+palabra.upper()`
	`print(frase)`
			B

C	`palabra='anaconda'`
	`txt='las vocales:'`
	`frase='Siempre '`
	`frase=frase+txt+palabra[2].upper()`
	`frase=frase+' y '+palabra[-4].upper()`
	`frase=frase+' están en '+palabra`
	`print(frase)`
		X	**C**

D	`palabra='anaconda'`
	`txt='las vocales:'`
	`frase='Siempre '`
	`frase=frase+txt+palabra[-3]`
	`frase=frase+' y '+palabra[2].upper()`
	`frase=' están en '+palabra`
	`print(frase)`
			D

Ejercicio 8 – Tema 3 **2 Ptos**

¿Cuál es la versión correcta de la función **edita**? El programa debe mostrar
ESTA<ESP>HISTORIA<ESP>CONTINUARÁ<ESP><ESP><ESP>

```
def edita(…):
  -
  -
  -

txt='esta historia continuará...'
print(edita(txt,',- .'))
```

Notas:
La función ***list()*** convierte un texto en una lista de caracteres
Ej:
txt='hola'
lis= list(txt) lis-> ['h', 'o', 'l', 'a']

El método ***join()*** genera una string con los caracteres que tiene como elementos una lista separándolos por el texto al inicio
Ej:
lis= ['h', 'o', 'l', 'a']
txt=''.join(lis)-> txt='h**o**l**a'**

A

```
def edita(carEsp):
    lista=list(t)
    for i in range(len(lista)):
        if lista[i] in carEsp:
            lista[i]='<esp>'
    t=''.join(lista)
    return lista
```

A

B

```
def edita(t,carEsp):
    lista=list(t)
    for i in range(len(lista)):
        if lista[i] in carEsp:
            lista[i]='<esp>'
    t=''.join(lista)
    return t.upper()
```

X

B

C

```
def edita(t,carEsp):
    lista=list(t)
    for i in range(len(lista)):
        if lista[i] in carEsp:
            lista[i]=''
    t=''.join(lista)
    return t
```

C

D

```
def edita(carEsp,t):
    lista=list(t)
    for i in range(len(lista)):
        if lista[i] in carEsp:
            lista[i]='<esp>'
    t=''.join(lista)
```

D

Ejercicio 9 – Tema 3	2 Ptos

¿Cuál condición es candidata a ser equivalente a la siguiente?

(num in sel) and (num!=int(num) or num>0 or num%3==0)

Atención!!
Te recomendamos que pruebes las condiciones con los siguientes valores:

```
sel=[2,21,-1,56]
num=22 y num=21
```

Nota:
Una condición (proposición) **p** es equivalente a otra **q**, si sus tablas de verdad coinciden. En otras palabras: si siempre que **p** da **V**, **q** da **V** y siempre que **p** da **F**, **q** da **F**

A	`(num in sel and num!=int(num) and num>0 and num%3==0)`		A
B	`num!=int(num) or num>0 or num%3==0`		B
C	`num in sel or num!=int(num) or num>0 or num%3==0`		C

D	`((num in sel and num!=int(num)) or`
	`(num>0 and num in sel) or`
	`(num in sel and num%3==0))`
		X	D

PENSAMIENTO COMPUTACIONAL (90) .UBAXXI

EXAMEN: PRIMER PARIAL – PRIMER CUATRIMESTRE 2025 **TEMA 1**

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	NOTA Y FIRMA DOCENTE (no rellenar)
TEL:	
AULA:	

Duración del examen: 1:30h.

- ✔ Escribir claramente el nombre en todas las páginas.
- ✔ El examen consta de 9 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 5 inclusive permiten acumular 1 punto (si son correctas), de la 6 a la 9 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**

Consignar con una X (cruz) la respuesta elegida en cada ejercicio en la siguiente matriz:

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Talón de Control para el Alumno (Tema 1)

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 2 Ptos	Ej 7 2 Ptos	Ej 8 2 Ptos	Ej 9 2 Ptos	
A										
B										
C										
D										

Ejercicio 1 – Tema 1			**1 Pto**
¿Cuál de los siguientes programas muestra por pantalla **barato**?			

A	`producto='clavo'`
	`unidad='caja'`
	`precio=275`
	`if producto in ('clavo','manguera','tornillo'):`
	`  if precio<1000 and unidad=='caja' and unidad=='kg':`
	`  print('carito')`
	`  elif precio<1000 and unidad=='unidad':`
	`  print('caro')`
	`elif producto in ('cola','pincel','lija'):`
	`  print('barato')`
			A

B	`producto='clavo'`
	`unidad='caja'`
	`precio=275`
	`if producto in ('manguera','tornillo'):`
	`  if precio>1000 and (unidad=='caja' or`
	`  unidad=='kg'):`
	`  print('carito')`
	`  elif precio<1000 and unidad=='unidad':`
	`  print('maso')`
	`elif producto in ('cola','lija','clavo'):`
	`  if unidad in ('ciento','docena','rollo'):`
	`  print('barato')`
	`  else:`
	`  print('caro')`
			B

C	`producto='clavo'`
	`unidad='caja'`
	`precio=275`
	`if producto in ('clavo','manguera','tornillo'):`
	`  if precio>1000 and`
	`  (unidad=='caja' or unidad=='kg'):`
	`  print('carito')`
	`  elif precio>1000 and unidad=='unidad':`
	`  print('caro')`
	`print('barato')`
		X	**C**

D	`producto='clavo'`
	`unidad='caja'`
	`precio=275`
	`if producto in ('clavo','manguera','tornillo'):`
	`  if precio>1000 or unidad=='caja'`
	`  or unidad=='kg':`
	`  print('carito')`
	`  elif precio>1000 and unidad=='unidad':`
	`  print('caro')`
	`else:`
	`  print('barato')`
			D

Ejercicio 2 – Tema 1 — **1 Pto**

¿Qué muestra el siguiente programa?

```
a=1
b=0
c=2.5
d=-2
print((a+d**2)/(c-b))
```

Nota:
El operador ** es potencia
Ej:
13 -> 1** y **5**2 -> 25**

A	`0.4`		**A**
B	`5.0`		**B**
C	`2.0`	**X**	**C**
D	`-4.0`		**D**

Ejercicio 3 – Tema 1 — **1 Pto**

¿Qué contenido tendrán al final las listas **a**, **b** y **c** en el siguiente programa?

```
indicador=[1,3,3,2,2,2,1]
numeros=[1,2,3,4,5,6,7]
a=[]
b=[]
c=[]
for i in range(len(numeros)):
    if indicador[i]==1:
        a.append(numeros[i])
    elif indicador[i]==2:
        b.append(numeros[i])
    else:
        c.append(numeros[i])
```

A	`a []`
	`b [1, 2, 3, 6, 7]`
	`c [4, 5]`
			A

B	`a [2, 7]`
	`b [1, 3]`
	`c [4, 5, 6]`
			B

C	`a [0, 0]`
	`b [2, 2]`
	`c [3, 3, 3]`
			C

D	`a [1, 7]`
	`b [4, 5, 6]`
	`c [2, 3]`
		X	**D**

Ejercicio 4 – Tema 1 | **1 Pto**

¿Cuál programa guarda en la lista **pares** únicamente los números pares y los almacena en **orden invertido** al original? En pares debe quedar **[22, 20]**
Recordá que un número es par si es mayor que cero y divisible por 2

Notas:
El método ***reverse()*** invierte una lista
Ej:
lis=[-1,2,0]
lis.reverse() lis-> [0,2,-1]

El método ***sort()*** ordena una lista
Ej:
lis=[6,-1,0]
lis.sort() -> lis=[-1, 0, 6]

El método ***insert(posición,nuevo elemento)*** permite agregar un elemento en una lista en cualquier posición
Ej:
lis=[1, 2, 3]
lis.insert(1,8) -> lis=[1, 8, 2, 3]

A

```
numeros=[15,-2,20,22,0,33,-1]
pares=[]
for num in numeros:
    if num>0 and num%2!=0:
        pares.append(num)
```

A

B

```
numeros=[15,-2,20,22,0,33,-1]
pares=[]
for num in numeros:
    if num>0 and num%2==0:
        pares.append(num)
pares.reverse()
```

X

B

C

```
numeros=[15,-2,20,22,0,33,-1]
pares=[]
for num in numeros:
    if num%2!=1:
        pares.insert(0,num)
```

C

D

```
numeros=[15,-2,20,22,0,33,-1]
pares=[]
for num in numeros:
    if num>0:
        pares.insert(0,num)
pares.sort()
```

D

Ejercicio 5 – Tema 1 — **1 Pto**

¿Qué devuelve la función **fun** en este programa?

```
def fun(t):
    vocales='AEIOUÁÉÍÓÚÜ'
    t=t.upper()
    cont=0
    for letra in t:
        if letra in vocales:
            cont+=1
    return cont>6

txt='Una oruga NEGRA en un PAJAR'
print(fun(txt))
```

A	11		**A**
B	0.0		**B**
C	True	**X**	**C**
D	False		**D**

Ejercicio 6 – Tema 1 — **2 Ptos**

¿Qué muestra por pantalla el siguiente programa?

```
cartel='BARCO'
extremo=''
centro=' '*(len(cartel)-2)
for i in range(len(cartel)//2):
  print(extremo+cartel[i]+centro+cartel[-(i+1)])
  extremo+=' '
  centro=centro[2:]
print(extremo+cartel[len(cartel)//2])
```

Notas:
Se puede seleccionar una porción de una string con **[desde:hasta]**, (recordá que **hasta** no entra)
Ej:
'hola, qué tal?'[4:] -> ', qué tal?'

El operador // es el cociente entero
Ej:
7//2 -> 3

El operador * aplicado a una string es repetición
Ej:
'hola'*3 -> 'holaholahola'

A	B O A C R	**X**	**A**
B	BARCO BARCO		**B**
C	bbbbb aaaaa rrrrr ccccc ooooo		**C**
D	B A R C O		**D**

Ejercicio 7 – Tema 1			2 Ptos
¿Cuál programa muestra el siguiente texto como salida? **Agustín, LA PLACA BACTERIANA DENTAL SE QUITA CON: PLAKOUT** **Nota:** Se puede seleccionar una porción de una string con **[desde:hasta]**, (recordá que **hasta** no entra) **Ej:** **'hola, qué tal?'[4:] -> ', qué tal?'**			

A	`mens='LA PLACA DENTAL SE QUITA CON'`
	`nom='Agustín'`
	`tipo='bacteriana'`
	`producto='plakout'`
	`textoMens=mens[:8]+' '+tipo`
	`textoMens=textoMens+mens[8:]`
	`textoMens=textoMens+' '+producto.upper()`
	`textoMens=textoMens+', '+nom.upper()`
	`print(textoMens)`
			A

B	`mens='LA PLACA DENTAL SE QUITA CON'`
	`nom='Agustín'`
	`tipo='bacteriana'`
	`producto='plakout'`
	`textoMens=nom+', '+mens[:8]+' '+tipo.upper()`
	`textoMens=textoMens+mens[8:]`
	`textoMens=textoMens+': '+producto.upper()`
	`print(textoMens)`
		X	**B**

C	`mens='LA PLACA DENTAL SE QUITA CON'`
	`nom='Agustín'`
	`tipo='bacteriana'`
	`producto='plakout'`
	`textoMens=mens[:8]+' '+tipo`
	`textoMens=mens[8:]`
	`textoMens=' '+producto`
	`textoMens=textoMens+', '+nom`
	`print(textoMens)`
			C

D	`mens='LA PLACA DENTAL SE QUITA CON'`
	`nom='Agustín'`
	`tipo='bacteriana'`
	`producto='plakout'`
	`textoMens=nom.upper()+', '+mens[8]+' '+tipo`
	`textoMens=mens[8]`
	`textoMens=textoMens+': '+producto.upper()`
	`print(textoMens)`
			D

Ejercicio 8 – Tema 1 **2 Ptos**

¿Cuál es la versión correcta de la función **sinLetras**? El programa debe mostrar
NTRRMNT N CDD GÓTC...

```
def sinLetras(…):
  -
  -
  -

txt='anteriormente en Ciudad Gótica...'
print(sinLetras(txt,'aeiou'))
```

Notas:
La función ***list()*** convierte un texto en una lista de caracteres
Ej:
txt='hola'
lis= list(txt) lis-> ['h', 'o', 'l', 'a']

El método ***join()*** genera una string con los caracteres que tiene como elementos una lista separándolos por el texto al inicio
Ej:
lis= ['h', 'o', 'l', 'a']
txt=''.join(lis)-> txt='h**o**l**a'**

El método ***remove()*** quita la primer ocurrencia de un elemento en una lista
Ej:
lis=[6,-1,0,5,6]
lis.remove(6) -> lis=[-1, 0, 5, 6]

A	`def sinLetras(t,let):`
	`  lista=list(t)`
	`  for car in let:`
	`  while car in lista:`
	`  lista.remove(car)`
	`  t=''.join(lista)`
	`  return t.upper()`
		X	A

B	`def sinLetras(t,let):`
	`  lista=list(t)`
	`  for car in let:`
	`  lista.remove(car)`
	`  t=''.join(lista)`
	`  return t`
			B

C	`def sinLetras():`
	`  lista=list(t)`
	`  for car in let:`
	`  while car in lista:`
	`  lista.remove(car)`
	`  return lista`
			C

D	`def sinLetras(t):`
	`  lista=list(t)`
	`  for car in let:`
	`  while car in lista:`
	`  lista.remove(car)`
	`  ''.join(lista)`
			D

Ejercicio 9 – Tema 1	2 Ptos

¿Cuál condición es candidata a ser equivalente a la siguiente?

(num>0 and num==int(num) and num%3==0) or (num in sel)

Atención!!
Te recomendamos que pruebes las condiciones con los siguientes valores:

```
sel=[2,25,-1,56]
num=22 y num=21
```

Nota:
Una condición (proposición) **p** es equivalente a otra **q**, si sus tablas de verdad coinciden. En otras palabras: si siempre que **p** da **V**, **q** da **V** y siempre que **p** da **F**, **q** da **F**

A	`num in sel or num==int(num) or num>0 or num%3<1`		A
B	`(num in sel and num==int(num) and num>0 and num%3<1)`		B
C	`((num in sel) or (not(num!=int(num)) and num>0 and num%3<1))`	X	C
D	`num not in sel or num==int(num)`		D