

PENSAMIENTO COMPUTACIONAL (90)

.UEAXI

TEMA 3

EXAMEN: EXAMEN JULIO 2025

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:

[illegible]

Talón de Control para el Alumno (Tema 3)

[illegible]

Ejercicio 1 – Tema 3		1 Pto	
¿Qué muestra por pantalla el siguiente programa? <pre>nivel='secundario' orientacion='química' materia='matemáticas' anio=4 if nivel not in ('secundario','terciario') or nivel[0]=='s': print('****') elif nivel in ('primario','secundario'): print('*') if materia in ('química orgánica','lengua','matemáticas'): print('*') if orientacion!='electrónica' and anio>=4: print('*') elif anio<3: print('*') else: print('***') print('*')</pre>			
A	*****		A
B	*** *	X	B
C	* **		C
D	* * * *		D

Ejercicio 2 – Tema 3		1 Pto	
¿Cuál programa produce la siguiente salida?			
4 es resultado de 2*2			
Nota: El método replace() permite reemplazar una substring (dentro de otra) por un texto diferente Ej: txt='hola, hola Chola !!' txt.replace('hola','chau') -> 'chau, chau Cchau !!'			
A	<pre>exp='CUATRO es dos x dos' digitos=('cero','uno','dos','tres','cuatro', 'cinco','seis','siete','ocho','nueve') exp=exp.upper() for i in range(len(exp)): if exp[i] in digitos: car=exp[i] exp=exp.replace(car,i) exp=exp.replace('*','X') exp=exp.replace('es','ES RESULTADO DE') print(exp)</pre>		A
B	<pre>exp='CUATRO es dos x dos' digitos=('cero','uno','dos','tres','cuatro', 'cinco','seis','siete','ocho','nueve') exp=exp.lower() for i in range(len(digitos)): if digitos[i] in exp: exp=exp.replace(digitos[i],str(i)) exp=exp.replace(' x ','*') exp=exp.replace('es','es resultado de') print(exp)</pre>	X	B
C	<pre>exp='CUATRO es dos x dos' digitos=('uno','dos','tres','cuatro', 'cinco','seis','siete','ocho', 'nueve','cero') for i in range(len(digitos)): if digitos[i] in exp: exp=exp.replace(digitos[i],str(i)) exp=exp.replace(' x ','*') exp=exp.replace('es','es resultado de') print(exp)</pre>		C
D	<pre>exp='CUATRO es dos x dos' digitos=('0','1','2','3','4', '5','6','7','8','9') exp=exp.lower() for i in range(len(digitos)): if digitos[i] in exp: exp=exp.replace(digitos[i],digitos[i].upper()) exp=exp.replace(' * ','x') exp=exp.replace('es','es resultado de') print(exp)</pre>		D

Ejercicio 3– Tema 3		1 Pto	
¿Cuál programa valida correctamente que se ingrese un horario (hora y minutos) de mañana correcta? Se considera mañana entre 0:1 y 12:0, ambos inclusive Debe insistir hasta conseguir un horario matutino válido			
A	<pre>pide=True while not pide: try: hora=int(input('Hora (0-12): ')) min=int(input('Minutos (0-59): ')) if ((hora==0 and min in range(1,60)) or (hora==12 and min==0) or (hora in range(1,12) and min in range(0,60))): else: print('Horario inválido') pide=False except ValueError: print('Deben ser valores enteros') print(hora,':',min)</pre>		A
B	<pre>pide=True while pide: try: hora=int(input('Hora (0-12): ')) min=int(input('Minutos (0-59): ')) if ((hora==0 and min in range(1,60)) or (hora==12 and min==0) or (hora in range(1,12) and min in range(0,60))): pide=False else: print('Horario inválido') except ValueError: print('Deben ser valores enteros') print(hora,':',min)</pre>	X	B
C	<pre>pide=False while not pide: hora=int(input('Hora (0-12): ')) min=int(input('Minutos (0-59): ')) if ((hora==0 and min in range(1,60)) or (hora==12 and min==0) or (hora in range(1,12) and min in range(0,60))): print('Ok') else: print('Horario inválido') try: print(hora,':',min) except ValueError: print('Deben ser valores enteros')</pre>		C
D	<pre>try: hora=int(input('Hora (0-12): ')) min=int(input('Minutos (0-59): ')) if (hora not in range(0,13) or (hora==12 and min>0) or (hora==0 and min==0) or min not in range(0,60)): print('Horario inválido') except ValueError: print('Deben ser valores enteros') print(hora,':',min)</pre>		D

Ejercicio 4 – Tema 3		1 Pto	
En el siguiente programa:			
<pre>def arma(b,stk): bSemanal=[] for tipo in b: cant=b[tipo] tipo=tipo.lower() for i in range(cant): bSemanal.append(stk[tipo][i]) return bSemanal bolsonTipo= ... #seleccionar la estructura adecuada stock={'tomates':['cherry','perita','redondo'], 'hojas':['mantecosa','espinaca','criolla'], 'zapallos':['zucchini','calabaza','cabutia']} bolson=arma(bolsonTipo,stock) print('Detalle del Bolsón de Verduras semanal') for verd in bolson: print(verd)</pre>			
Cuál es la definición correcta de la estructura <i>bolsonTipo</i> para que funcione adecuadamente y genere la siguiente salida:			
Detalle del Bolsón de Verduras semanal mantecosa espinaca zucchini calabaza cherry			
A	bolsonTipo=[2,2,1]		A
B	bolsonTipo={'HOJAS':2,'ZAPALLOS':2,'TOMATES':1}	X	B
C	bolsonTipo={2:['HOJAS','ZAPALLOS'], 1:['TOMATES']}		C
D	bolsonTipo={'tomates':1,['cherry','perita','redondo'], 'hojas':2,['mantecosa','espinaca','criolla'], 'zapallos':2,['zucchini','calabaza','cabutia']}		D

Ejercicio 5 – Tema 3			1 Pto
¿Qué muestra por pantalla el siguiente programa? <pre>colores=['rosa','azul','amarillo','verde'] tonos=['claro','oscuro','medio'] tonalidades=[] for i in range(len(colores)): val=-(i%3+1) tonalidades.append(colores[-(i+1)]) tonalidades.append(tonos[val]) print(tonalidades)</pre> Nota: Se puede seleccionar elementos de una lista de atrás para adelante con un índice negativo Ej: lis=['a','b','c'] lis[-2] -> 'b'			
A	['verde','amarillo','azul','rosa','medio','oscuro','claro','medio']		A
B	[['verde','medio'],['amarillo','oscuro'], ['azul','claro'], ['rosa','medio']]		B
C	['verde','claro','amarillo','oscuro','azul','medio']		C
D	['verde','medio','amarillo','oscuro','azul','claro','rosa','medio']	X	D

Ejercicio 6 – Tema 3			2 Ptos
Para el siguiente DataFrame <i>mundialClubes</i> <pre> nomClub pts partJug octavos 0 Atlético Madrid 0 3 False 1 Bayern Munich 3 3 False 2 Benfica 0 1 False 3 Boca Juniors 2 2 True 4 Botafogo 0 2 True 5 Chelsea 3 3 False 6 Inter Milan 1 1 True 7 Inter Miami 2 2 True 8 Manchester City 3 3 True 9 Paris Saint-Germain 9 3 False 10 Real Madrid 3 3 False 11 River Plate 3 3 True</pre> ¿Qué instrucción produce la siguiente salida? <pre> nomClub octavos 0 Atlético Madrid False 1 Bayern Munich False 2 Benfica False 3 Boca Juniors True 4 Botafogo True 5 Chelsea False 6 Inter Milan True 7 Inter Miami True 8 Manchester City True 9 Paris Saint-Germain False 10 Real Madrid False 11 River Plate True</pre>			
A	print(mundialClubes[['nomClub','octavos']])	X	A
B	print(mundialClubes[['pts','octavos']])		B
C	print(mundialClubes.tail(3))		C
D	print(mundialClubes.head(6))		D

Ejercicio 7 – Tema 3		2 Ptos	
¿Qué versión de la función edita() será la adecuada para eliminar los acentos?			
<pre>def edita (...): - - - txt='condición: cúrcuma con anís es más fácil' vocCon='áéíóú' vocSin='aeiou' print(edita(txt,vocCon,vocSin))</pre>			
La salida debe ser: CONDICION: CURCUMA CON ANIS ES MAS FACIL			
Nota: El método replace() permite cambiar todas las ocurrencias de la primera substring por la segunda Ej: txt='sala de la cAsa' txt=txt.replace('a','e') txt -> 'sele de le cAse'			
A	<pre>def edita(t,sin,con): t=t.lower() for i in range(len(con)): t.replace(con[i],sin[i]) return t</pre>		A
B	<pre>def edita(t): t=t.lower() for i in range(len(con)): t=t.replace(con[i],sin[i])</pre>		B
C	<pre>def edita(t,con,sin): t=t.lower() for i in range(len(con)): t=t.replace(con[i],sin[i]) return t.upper()</pre>	X	C
D	<pre>def edita(t,con,sin): for i in range(len(con)): t.replace(con[i],sin) return t.lower()</pre>		D

Ejercicio 8– Tema 3

2 Ptos

¿Qué salida produce el siguiente programa?

```
def edita(naip):
    n1=naip[0][0].upper()+' de '
    n2=naip[1][0].upper()+' de '
    n3=naip[2][0].upper()+' de '
    return [(n1,naip[0][1]),(n2,naip[1][1]),(n3,naip[2][1])]

def negras(naipes):
    neg=('REY de ','CABALLO de ','SOTA de ')
    return naipes[0][0] in neg or naipes[1][0] in neg or
           naipes[2][0] in neg

mano=[(('2','Oros'),('3','Oros'),('7','Copas')),
       (('2','Espadas'),('4','Oros'),('5','Copas')),
       (('Rey','Copas'),('4','Copas'),('6','Copas')),
       (('7','Espadas'),('Caballo','Espadas'),('2','Bastos')]]
naipes=list(map(edita,mano))
tieneNegras=list(filter(negras,naipes))
if len(tieneNegras)>0:
    print('Hay NEGRAS en la mano!')
    for naipes in tieneNegras:
        salida=naipes[0][0]+naipes[0][1]+' - '
        salida+=naipes[1][0]+naipes[1][1]+' - '
        salida+=naipes[2][0]+naipes[2][1]
    print(salida)
```

A	Hay NEGRAS en la mano!		A
B	Hay NEGRAS en la mano! 2 de Oros - 3 de Oros - 7 de Copas 2 de Espadas - 4 de Oros - 5 de Copas Rey de Copas - 4 de Copas - 6 de Copas 7 de Espadas - Caballo de Espadas - 2 de Bastos		B
C	Hay NEGRAS en la mano! REY de Copas - 4 de Copas - 6 de Copas 7 de Espadas - CABALLO de Espadas - 2 de Bastos	X	C
D	Hay NEGRAS en la mano! 2 de Oros - 3 de Oros - 7 de Copas 2 de Espadas - 4 de Oros - 5 de Copas		D

Ejercicio 9– Tema 3		2 Ptos	
¿Qué contendrá el archivo datosAlumnos.txt al final de este programa? El archivo inicialmente no existe			
<pre>def guarda (d,nom,dcc,a): linea=str(d)+' - '+nom[d][1].upper()+' - '+nom[d][0]+'\\n' a.write(linea) linea=str(d)+' - '+dcc[d][0].upper()+' - '+dcc[d][1]+'\\n' a.write(linea) direcciones={45667718:['Condarco 5325','gabiorg@org.com'], 42164671:['Medrano 1205 4to B','tamiorg@org.com'], 43301718:['Caracas 3532 2do A','lucasorg@org.com'], 39002166:['Malabia 1316 3ro A','anaorg@org.com']} nombres={39002166:['Ana Inés','Lempert'], 45667718:['Gabriel','Urriza'], 43301718:['Lucas Iván','Pérez Cort'], 42164671:['Tamara','Tenti']} arch=open('datosAlumnos.txt','w') for dni in nombres: guarda(dni,nombres, direcciones,arch) arch.close()</pre>			
A	39002166 - LEMPERT - Ana Inés 39002166 - MALABIA 1316 3RO A - anaorg@org.com 45667718 - URRIZA - Gabriel 45667718 - CONDARCO 5325 - gabiorg@org.com 43301718 - PÉREZ CORT - Lucas Iván 43301718 - CARACAS 3532 2DO A - lucasorg@org.com 42164671 - TENTI - Tamara 42164671 - MEDRANO 1205 4TO B - tamiorg@org.com	X	A
B	45667718 - URRIZA - Gabriel 45667718 - CONDARCO 5325 - gabiorg@org.com 42164671 - TENTI - Tamara 42164671 - MEDRANO 1205 4TO B - tamiorg@org.com		B
C	42164671 - MEDRANO 1205 4TO B - tamiorg@org.com		C
D	39002166 - Ana Inés - Lempert;Malabia 1316 3ro A - anaorg@org.com 43301718 - Lucas Iván - Pérez Cort - Caracas 3532 2do A - lucasorg@org.com		D