

PENSAMIENTO COMPUTACIONAL (90)

UBAXXI

TEMA 5

EXAMEN: PRIMER PARCIAL

| | |
|----------------------------------|---------------------------------------|
| 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 10 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 7 inclusive permiten acumular 1 punto (si son correctas), de la 8 a la 10 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 |

Matriz de Respuestas

| | | | | | | | | | | |
|----------|---------------|---------------|---------------|---------------|---------------|----------------|----------------|----------------|----------------|----------|
| | 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 | |
| 1 | | | | | | | | | | 1 |
| 2 | | | | | | | | | | 2 |
| 3 | | | | | | | | | | 3 |
| 4 | | | | | | | | | | 4 |

¡ATENCIÓN! Las respuestas sólo se considerarán válidas si se encuentran en la matriz. De haber diferencias entre la opción seleccionada en el ejercicio y en la matriz, se considerará como válida esta última.

Talón de Control para el Alumno

| | | | | | | | | | | |
|----------|---------------|---------------|---------------|---------------|---------------|----------------|----------------|----------------|----------------|----------|
| | 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 | |
| 1 | | | | | | | | | | 1 |
| 2 | | | | | | | | | | 2 |
| 3 | | | | | | | | | | 3 |
| 4 | | | | | | | | | | 4 |

| | | | |
|--|--------------|----------|----------|
| 0105 – 1 Pto | | | |
| ¿Qué resultado muestra el siguiente programa? | | | |
| <pre>a=2 b=0 c=2.5 d=3.5 print(d-c*a+int(c)*(b+a))</pre> | | | |
| 1 | 8 | | 1 |
| 2 | -18.5 | | 2 |
| 3 | 2.5 | X | 3 |
| 4 | 0 | | 4 |

| 0205 – 1 Pto | | | |
|--|--|---|---|
| ¿Qué programa evalúa menos condiciones en total? | | | |
| 1 | <pre> categoria='botón' clase='metal' color='plata' if categoria not in ('hilo','botón','cinta'): if clase=='plástico' and color=='azul': print('Cierre') if clase=='gancho': print('Pasador') if categoria=='botón': if color=='azul' or color=='plata': print('Tengo') if categoria=='hilo': print('Varios colores') if categoria=='cinta': print('raso o seda') </pre> | | 1 |
| 2 | <pre> categoria='botón' clase='metal' color='plata' if categoria in ('hilo','cinta','botón'): if color in ('rojo','blanco','gris'): print('Tengo') elif clase=='metal': if color=='oro': print('No tengo') else: print('Tengo') elif categoria=='cinta': if color in ('oro','plata','peltr'): print('Hay') elif categoria=='hilo': print('raso o seda') </pre> | | 2 |
| 3 | <pre> categoria='botón' clase='metal' color='plata' if categoria in ('hilo','cinta'): if clase=='seda': print('Tengo') if categoria=='aguja': print('fina o gruesa?') if categoria=='botón': if color=='azul' or color=='plata': print('Tengo') if categoria=='cinta': if color in ('oro','plata','peltr'): print('Hay') elif categoria=='hilo': print('raso o seda') </pre> | | 3 |
| 4 | <pre> categoria='botón' clase='metal' color='plata' if categoria not in ('hilo','botón','cinta'): if clase=='plástico' and color=='azul': print('Cierre') if clase=='gancho': print('Pasador') else: print('Elige color') </pre> | X | 4 |

| 0305 – 1 Pto | | | |
|--|--|---|---|
| ¿Qué muestra por pantalla el siguiente programa? | | | |
| <pre> pares=[12,222,4,8,2] ordenSerie=['','Inicial','bajo', 'bajo','bajo','media','media','uff!'] print('Zona en la serie de los pares') for num in pares: orden=num//2 if orden>len(ordenSerie)-1: orden=len(ordenSerie)-1 print(num,ordenSerie[orden]) </pre> | | | |
| 1 | Zona en la serie de los pares 12 222 4 8 2 | | 1 |
| 2 | Zona en la serie de los pares 12 uff! 222 uff! 4 uff! 8 uff! 2 uff! | | 2 |
| 3 | Zona en la serie de los pares 12 media 222 uff! 4 bajo 8 bajo 2 Inicial | X | 3 |
| 4 | Zona en la serie de los pares Inicial bajo bajo uff! media | | 4 |

| | | | |
|---|------------------------------|---|---|
| 0405 – 1 Pto | | | |
| <p>¿Qué muestra por pantalla el siguiente programa?</p> <pre>nom1='Juan' nom2='Elena' nom2=nom2.lower() texto='Mis primos son:' print(texto) texto=nom2[:2]+' - '+nom1[-1] print(texto)</pre> <p>Notas: Recordá que los caracteres de las string se numeran de 0 en adelante Se puede seccionar una string con [:] Ejs 'marca'[2:] -> 'rca' - 'marca':[3] -> 'mar' y 'marca'[1:4] -> 'arc'</p> <p>Se puede invertir una string con el siguiente tip a[::-1] Ej 'hola'[::-1] -> 'aloh'</p> <p>Se puede referenciar un elemento (carácter) con un índice positivo o negativo. Positivo toma el n elemento desde la izquierda; negativo toma el n elemento desde la derecha (antes del final) Ejs 'notas'[1] -> 'o' y 'notas'[-1] -> 's' o 'notas'[-3] -> 't'</p> | | | |
| 1 | Mis primos son: el - JUAN | | 1 |
| 2 | Mis primos son: el - n | X | 2 |
| 3 | Mis primos son: elena - juan | | 3 |
| 4 | Elena Juan | | 4 |

| | | | |
|---|-------------------|---|---|
| 0505 – 1 Pto | | | |
| <p>¿Qué muestra por pantalla el siguiente programa?</p> <pre>pal='CALOR' for i in range((len(pal)//2)+1): corre=' *(i+1) muestra=pal[i:len(pal)-i] print(corre+muestra)</pre> <p>Notas: El operador // es el cociente entero Ej 7//2->3 a diferencia de 7/2->3.5</p> <p>El operador * repite n veces una string Ej 'una'*2 -> 'unauna'</p> <p>Se puede seccionar una string con [:] Ejs 'marca'[2:] -> 'rca' - 'marca':[3] -> 'mar' y 'marca'[1:4] -> 'arc'</p> | | | |
| 1 | L ALO CALOR | | 1 |
| 2 | CALOR CALOR | | 2 |
| 3 | CALOR ALO L | X | 3 |
| 4 | LORCA | | 4 |

| 0605 – 2 Ptos | | | |
|--|------------------------------------|---|---|
| ¿Qué muestra el siguiente programa? | | | |
| <pre> a=[4,6,0,2] b=['ana','pedro','luis','emilia','li'] c=[] for num in a: c.append(len(b[num//2])) print(c) </pre> | | | |
| Nota: | | | |
| El operador // es el cociente entero | | | |
| Ej | | | |
| 7//2->3 a diferencia de 7/2->3.5 | | | |
| 1 | [4, 4] | | 1 |
| 2 | ['LUIS', 'EMILIA', 'ANA', 'PEDRO'] | | 2 |
| 3 | [4, 6, 3, 5] | X | 3 |
| 4 | [2, 0, 6] | | 4 |

| 0705 – 2 Ptos | | | |
|--|---|---|---|
| ¿Qué programa ejecuta menos veces el cuerpo del bucle? | | | |
| 1 | <pre> i=1 j=1 while i>j: i-=1 </pre> | X | 1 |
| 2 | <pre> for car in 'mamarracho': i=0 </pre> | | 2 |
| 3 | <pre> for i in range(3): j=1 </pre> | | 3 |
| 4 | <pre> i=1 j=5 while i<=j: i+=1 </pre> | | 4 |

| 0805 – 2 Ptos | | | |
|--|---|---|---|
| <p>¿Qué función es adecuada para el siguiente programa?</p> <pre>def calcula (...): - - - term1=[5,0,3] term2=[3,6,2] for i in range(len(term2)): print(calcula(term2[i],term1[i]))</pre> <p>Debe mostrar la siguiente salida:</p> <p>8 6 5</p> | | | |
| 1 | <pre>def calcula(): resultado=n1-n2 return resultado</pre> | | 1 |
| 2 | <pre>def calcula(n1,n2): resultado=n1+n2 return resultado</pre> | X | 2 |
| 3 | <pre>def calcula(n1,n2): resultado=n1-1 return 1</pre> | | 3 |
| 4 | <pre>def calcula(n2): resultado=n1+n2</pre> | | 4 |

| 0905 – 2 Ptos | | | |
|--|--|---|---|
| <p>¿Qué muestra por pantalla el siguiente programa?</p> <pre>num=0 calc=1 for i in range(1,8,3): num+=i print('@'*num) while num>10: print('@'*(num-10)) num-=1</pre> <p>Nota: El operador * aplicado a una string repite n veces la misma</p> <p>Ej: a='mi'*2 -> a='mimi'</p> | | | |
| 1 | @ @ @ @ @ | | 1 |
| 2 | @ @@@@@ @@@@@@@@@@@@@@@ @@ @ | X | 2 |
| 3 | @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ | | 3 |
| 4 | @ @@ @@@@@@@ @@@@@ @ | | 4 |

