

<b>Comenzado el</b>	miércoles, 29 de marzo de 2023, 20:08
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	miércoles, 29 de marzo de 2023, 20:49
<b>Tiempo empleado</b>	41 minutos 11 segundos
<b>Puntos</b>	28/31
<b>Calificación</b>	9 de 10 (89%)

Pregunta **1**

Correcta

Se puntúa 1 sobre 1

¿Cuál de las siguientes afirmaciones es cierta respecto de un *diagrama de flujo*?

Seleccione una:

- ☐ a. En un diagrama de flujo deben ponerse hasta los detalles mínimos: colores usados en la pantalla, mensajes aclaratorios en pantalla, comas y símbolos específicos de la sintaxis de un lenguaje, etc.
- ☐ b. Se hace un diagrama de flujo para cada lenguaje en que se vaya a programar, aunque el problema sea siempre el mismo.
- ☐ c. Un diagrama es un gráfico que los profesores inventaron para torturar a los sufridos y nunca bien comprendidos alumnos.
- ☒ d. Es un gráfico que permite ver claramente la lógica de un algoritmo, sin entrar en los detalles de la sintaxis de un lenguaje ✓ ¡Ok!

¡Correcto!

Pregunta **2**

Correcta

Se puntúa 1 sobre 1

¿Qué es en programación una *Estructura Secuencial de Instrucciones*?

Seleccione una:

- ☒ a. Un bloque de instrucciones simples (asignaciones, visualizaciones, lecturas) escritas una debajo de la otra ✓ Ok.
- ☐ b. Un conjunto de  $n$  variables del mismo tipo.
- ☐ c. Un bloque de comentarios de texto incluido en un programa.
- ☐ d. Una forma de organizar un conjunto de  $n$  datos para facilitar su acceso desde un programa.

¡Correcto!

## Pregunta 3

Correcta

Se puntúa 2 sobre 2

¿Cuáles de las siguientes expresiones Python calculan la **raíz cuarta** del valor contenido en la variable **a**? (Suponga que **a** contiene un número positivo) (Más de una respuesta puede ser correcta, por lo que asegúrese de marcar todas las que considere válidas).

Seleccione una o más de una:

- ☒ a. `r1 = pow(a, (1/4))` ✓ ¡Correcto! La raíz cuarta de **a** se puede escribir como  $a^{(1/4)}$  que en Python puede escribirse como **pow(a, (1/4))**.
- ☒ b. `r1 = a ** 0.25` ✓ ¡Correcto! La raíz cuarta de **a** se puede expresar como  $a^{(1/4)}$  que es lo mismo que  $a^{0.25}$  o sea: **a\*\*0.25** en Python.
- ☐ c. `r1 = pow(a, -4)`
- ☐ d. `r1 = pow(0.25, a)`
- ☐ e. `r1 = a ** 4`
- ☒ f. `r1 = a ** (1/4)` ✓ ¡Correcto! Si el exponente es 1/4, se está calculando la raíz cuarta.

¡Correcto!

## Pregunta 4

Correcta

Se puntúa 1 sobre 1

¿Cuál de las siguientes afirmaciones es **falsa** respecto de la técnica de *pseudocódigo* para representación de algoritmos?

Seleccione una:

- ☐ a. El programador es quien decide la forma, la profundidad y el nivel de detalle expresado en la lógica y en la estructura de un algoritmo.
- ☒ b. El planteo de un esquema de pseudocódigo se realiza siempre en base a reglas y estándares estrictos que los programadores deben conocer y respetar. ✓ ¡Ok! Recuerde: se pidió indicar cuál de las consignas es FALSA... y esto es efectivamente falso en cuanto al planteo de pseudocódigos.
- ☐ c. El pseudocódigo está pensado para ser leído e interpretado por una persona, y no por una computadora.
- ☐ d. El pseudocódigo puede basarse (con menor o mayor rigor) en la estructura general de un lenguaje particular, y en este caso se designa como un *pseudocódigo estructurado*.

¡Correcto!

## Pregunta 5


Correcta

Se puntúa 2 sobre 2

¿Qué efecto produce el siguiente script en Python?

```
b = 20
B = None
y = b * 3
print("Valor de y: ", y, "Valor de b:", b)
print("Valor de b: ", B)
```

Seleccione una:

- ☐ a. Produce un error al intentar ejecutar la última línea (la variable *B* no está definida).
- ☒ b. Ejecuta sin problemas: queda *b = 20*, *B = None*, *y = 60*, Y  ¡Ok! Las variables *b* y *B* son diferentes, y no hay problema en asignar el valor *None* a una variable.
- ☐ c. Ejecuta sin problemas: queda *b = None*, *y = None*, *B = None*.
- ☐ d. Produce un error al intentar ejecutar la tercera línea: *y = b \* 3* (no se puede usar el valor *None* como numérico)

¡Correcto!

## Pregunta 6

Correcta


Se puntúa 1 sobre 1

Suponga que está trabajando directamente con el editor del *shell de Python* (por ejemplo, a través del *IDLE GUI*). ¿Qué efecto producirá el siguiente script? (aclaración: los símbolos ">>>" conforman el prompt del IDLE, y **no deben ser escritos por el programador**... sólo escriba las instrucciones que se marcan abajo en **color azul**):

```
>>>var = 12
>>>var
```

Observación: también suponemos que el programador escribirá el script *línea por línea presionando <Enter>* al final de cada una en el shell, y **NO** que hará "*copy & paste*" de este bloque en el shell.

Seleccione una:

- ☐ a. Provocará un error de ejecución en la segunda línea.
- ☐ b. Ejecutará correctamente, pero no mostrará nada en la consola de salida.
- ☒ c. Mostrará el valor de la variable *var* (un  Correcto. Si está trabajando directamente con el editor del shell, no es necesario usar la función *print()* para visualizar el valor de una variable...
- ☐ d. Ejecutará correctamente, pero mostrará en consola de salida el valor *None*.

¡Correcto!

Pregunta **7**

Correcta

Se puntúa 2 sobre 2

¿Qué valor queda valiendo la variable **x**, luego de la siguiente secuencia de instrucciones en *Python 3*?

```
p = 2
x = 20
x = p * 8
x = p + 1
x = 17 + p
x = int((input('Ingrese un número entero: ')))
```

Seleccione una:

- ☒ a. Quedará valiendo el número cargado por teclado en la última línea.
- ☐ b. 19
- ☐ c. 16
- ☐ d. 20

¡Correcto! Cualquier otro valor que **x** haya tomado antes de esa línea se perderá (por supuesto, el usuario podría cargar un valor igual a alguno que **x** tuviese anteriormente, pero eso no es predecible en este contexto).

¡Correcto!

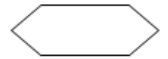
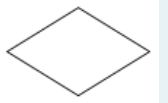


Pregunta **8**

Correcta

Se puntúa 1 sobre 1

¿Cuál de los siguientes símbolos NO representa un *proceso* en un *diagrama de flujo*?

Seleccione una:

- ☐ a. 
- ☐ b. 
- ☒ c. 
- ☐ d. 

¡Ok! Efectivamente, este símbolo representa una operación de carga de datos y no un proceso de transformación de datos en resultados.

¡Correcto!

Pregunta 9

Correcta

Se puntúa 2 sobre 2

Analice el siguiente script simple, cuyo objetivo es tomar por teclado los datos básicos de un estudiante, y mostrar por consola estándar los datos cargados:

```
n = input('Ingrese su nombre: ')
print('El nombre ingresado es: ', n)
n = int(input('Ahora ingrese su número de legajo: '))
print('El legajo ingresado es: ', n)
n = float( input('Y ahora ingrese su promedio: ') )
print('El promedio ingresado es: ', n)
```

¿Producirá algún problema la ejecución de este script?

Seleccione una:

- ☐ a. Sí. El script ejecutará sin problemas aparentes, pero mostrará en consola estándar siempre el mismo valor: el valor *None*.
- ☐ b. Sí. El script ejecutará sin problemas aparentes, pero mostrará en consola estándar siempre el mismo valor: el nombre del estudiante.
- ☒ c. No. El script se ejecutará sin problemas y hará lo esperado. ✓ ¡Correcto!
- ☐ d. Sí. El script comenzará a ejecutarse, pero lanzará un error y se interrumpirá cuando intente cargar el legajo del estudiante en la variable **n** que ya contenía el nombre.

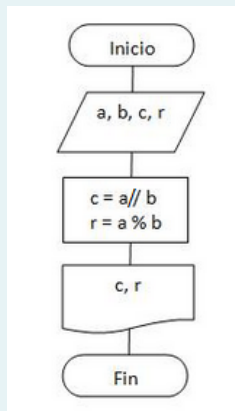
¡Correcto!

Pregunta **10**

Correcta

Se puntúa 1 sobre 1

Suponga que se desea desarrollar un programa que cargue dos números, y muestre el cociente entero y resto de la división entre esos dos números. A continuación se muestra el diagrama de flujo propuesto y el programa en Python. ¿Está bien planteado el diagrama de flujo?



```
# script en Python...
a = int(input('A: '))
b = int(input('B: '))

c = a // b
d = a % b

print('Cociente:', c)
print('Resto:', d)
```

Seleccione una:

- ☐ a. El diagrama está mal planteado (y también el programa): debió usarse una condición para verificar primero si el divisor (*b*) es menor o igual al dividendo (*a*).
- ☐ b. El diagrama está mal planteado: el símbolo usado al final para indicar visualización de resultados, debió ser un rectángulo.
- ☒ c. El diagrama está mal planteado: en el símbolo de carga por teclado (el paralelogramo) está indicando la *carga* de las ¡Ok!
- ☐ d. Sí. El diagrama está correctamente planteado.

¡Correcto!

Pregunta **11**

Correcta

Se puntúa 2 sobre 2

Suponga que las variables  $a$ ,  $b$  y  $c$  están correctamente asignadas con números en forma previa, y considere la siguiente expresión en Python:

```
d = (a + b) + c // a - (b - c)
```

¿Cuál de las siguientes es equivalente a la expresión anterior (cuál de ellas obtiene idéntico resultado para la variable  $d$ )?

Seleccione una:

- ☐ 1.  $d = ((a + b + c) // a) - b - c$
- ☐ 2.  $d = a + b + c // (a - b) - c$
- ☐ 3.  $d = (a + b) + c // (a - b - c)$
- ☒ 4.  $d = (a + b) + (c // a) - (b - c)$  ✓ ¡Correcto!

¡Correcto!

Pregunta **12**

Correcta

Se puntúa 1 sobre 1

Sabemos que un *IDE* es un programa que provee herramientas para editar, depurar y ejecutar con sencillez y eficiencia un programa desarrollado en algún lenguaje de programación. El *IDE* que usaremos a lo largo del curso es el *PyCharm Edu*. Concretamente, ¿qué significa la sigla *IDE*?

Seleccione una:

- ☐ a. Integrated Development Engine (Motor Integrado de Desarrollo)
- ☒ b. Integrated Development Environment (Entorno Integrado de Desarrollo) ✓ Ok.
- ☐ c. Integrated Database Engine (Motor Integrado de Bases de Datos)
- ☐ d. Integrated Database Environment (Entorno Integrado de Bases de Datos)

¡Correcto!

Pregunta **13**

Correcta

Se puntúa 2 sobre 2

¿Hay algún inconveniente en el script Python que sigue?

```
a = int(input('Primer número: '))
    b = int(input('Segundo número: '))
c = a + b
    print('Suma:', c)
```

Seleccione una:

- ☐ a. No. No hay ningún problema.
- ☒ b. Sí. Está mal indentado en general (líneas que pertenecen al mismo bloque de instrucciones están en columnas diferentes), y provocará un error de intérprete. ¡Correcto!
- ☐ c. Sí. Están mal usadas las funciones input() e int() (no pueden combinarse en la forma mostrada en el script).
- ☐ d. Sí. Al cargar los datos, estos se están ingresando y asignando como cadenas de caracteres, de modo que al hacer la suma el resultado será la concatenación de las cadenas en lugar de la suma de los números esperados.

¡Correcto!

Pregunta **14**

Correcta

Se puntúa 1 sobre 1

Suponga que tiene un script Python almacenado en un archivo llamado "*script.py*". Suponga también que está trabajando bajo sistema operativo Windows, y que el archivo *script.py* está guardado en la carpeta "*C:\Programas*". Finalmente, suponga que la variable de entorno *PATH* de Windows contiene correctamente la ruta del intérprete de Python. En estas condiciones, ¿cuál de las siguientes órdenes provocará que el script sea ejecutado desde la línea de órdenes de Windows, asumiendo que la carpeta activa es la que indica en cada caso el prompt?

Seleccione una:

- ☐ a. C:\Programas>python script
- ☐ b. C:\Programas>python prueba.py
- ☒ c. C:\Programas>python script.py ¡Ok!
- ☐ d. C:\Program Files>python script.py

¡Correcto!



Pregunta **15**

Correcta

Se puntúa 2 sobre 2

¿Cuál es el valor final de la variable **res**, luego de aplicar la siguiente secuencia de instrucciones en Python? (Es recomendable que primero intente ejecutar este script y *luego* conteste a esta pregunta):

```
a = 20
b = 6
res = ((a * b) // 4) % 7
```

Respuesta:

2



¡Correcto!

Pregunta **16**

Correcta

Se puntúa 1 sobre 1

¿Cuántos números enteros diferentes pueden representarse *en binario* si se dispone de un conjunto de 8(ocho) **bytes** agrupados? (Si... dice **8 bytes**, NO 8 bits...)

Seleccione una:

- ☐ a. **64<sup>2</sup>** (= 4096 números enteros diferentes)
- ☐ b. **2<sup>8</sup>** (= 256 números enteros diferentes)
- ☒ c. **2<sup>64</sup>**  
(= 18446744073709551616 números enteros diferentes)
- ☐ d. **2<sup>16</sup>** (= 65536 números enteros diferentes)



¡Ok! Cada bit permite 2 combinaciones, y dispone de (8 bytes \* 8 bits cada uno) = **64** bits en total, por lo que la cantidad total de combinaciones (o sea, la cantidad de números diferentes que pueden representarse) es 2<sup>64</sup>.

¡Correcto!

Pregunta **17**

Correcta

Se puntúa 2 sobre 2

¿Qué hace el siguiente script en Python?

```
__author__ = 'Cátedra de AED'

c1 = float(input('Ingrese el primer valor: '))
c2 = float(input('Ingrese el segundo valor: '))
c3 = float(input('Ingrese el tercer valor: '))

res = c1 + c2 + c3 / 3
print('Resultado:', res)
```

Seleccione una:

- ☐ a. Calcula y muestra el cociente entre el valor **c3** y el número **3**.
- ☒ b. Calcula y muestra el resultado de hacer **c1 + c2 + (c3 / 3)**. ✓ ¡Correcto!
- ☐ c. Calcula y muestra el promedio real de los valores **c1**, **c2** y **c3**.
- ☐ d. Calcula y muestra el porcentaje que el valor **c1** representa sobre el total **c1 + c2 + c3**.

¡Correcto!

Pregunta **18**

Parcialmente correcta

Se puntúa 1 sobre 2

¿Cuáles de las siguientes son propiedades básicas del resto de una división (y por lo tanto, aplicables al *operador resto o módulo* en un lenguaje de programación)?

**Observación:** note que **MAS DE UNA** respuesta puede ser correcta. Marque **TODAS** las que considere válidas.

Seleccione una o más de una:

- ☒ a. Si se divide un número entero positivo **x** por otro número entero positivo **n**, los posibles restos son todos los números en el intervalo **[0, n-1]** (y serían entonces, **n** posibles valores distintos). ✓ ¡Ok! Se deduce de la propia definición del resto de una división entre números enteros positivos.
- ☒ b. Si el resto de dividir un número **x** por otro **n** es cero, entonces **x** es múltiplo de **n** (o lo que es lo mismo, **x** es divisible por **n**). ✓ ¡Ok!
- ☐ c. El resto de dividir un número **x** por otro **n**, puede ser igual al número **x**.
- ☒ d. El resto de dividir un número entero positivo **x** por otro entero positivo **n**, puede ser un número mayor a **n**. ✗ Incorrecto... el resto no puede ser mayor que el dividendo... pues la división continuaría en ese caso...

¡Correcto!

Pregunta **19**

Incorrecta

Se puntúa 0 sobre 2

Suponga que el Despacho de Alumnos de una Universidad cuenta con 10 oficinas numeradas en forma correlativa entre 1 y 10. Cada estudiante que llega a realizar un trámite debe ser enviado a una de las diez oficinas y para determinar el número de la oficina se usa como dato el número de *legajo* del estudiante. Suponiendo que el número de legajo está almacenado correctamente en la variable *leg*, ¿cuál de las siguientes expresiones calculará correctamente el número de la oficina donde debe enviarse a cada estudiante?

Seleccione una:

☐ a.  $\text{oficina} = \text{leg} \% 10 + 1$ 

☒ b.  $\text{oficina} = \text{leg} \% 10$  **Incorrecto...** el cálculo  $\text{leg} \% 10$  entrega un valor que siempre estará en el intervalo  $[0, 9]$ ... pero las oficinas están numeradas con valores del intervalo  $[1, 10]$ ... ¿Qué hará un estudiante que sea enviado a la oficina 0 que ni siquiera existe?

☐ c.  $\text{oficina} = \text{leg} \% 7 + 1$ ☐ d.  $\text{oficina} = \text{leg} \% (7 + 1)$ **Incorrecto...** revise la Ficha 02 (página 43 en adelante)Pregunta **20**

Correcta

Se puntúa 2 sobre 2

¿Cuál de los siguientes conjuntos **NO** es la caracterización de una clase de congruencia (módulo 4)? (Recuerde que se denota como  $\mathbb{Z}$  al conjunto de los números enteros) (**LEA** la sección 5 de la Ficha 02, y **haga un esfuerzo por intentar comprender** lo que allí se expone, **antes** de responder esta pregunta o de ir al foro para pedir que le ayuden a entender...)

Seleccione una:

☐ a.  $\{4*k + 1 \mid (\forall k \in \mathbb{Z})\}$ 

☒ b.  $\{5*k + 1 \mid (\forall k \in \mathbb{Z})\}$  **✓** ¡Ok! Efectivamente, los números que caracterizan a este conjunto son de la forma  $5*k + 1$ , lo cual indica que todos ellos dejan un resto de 1 pero al dividir por 5 (y no necesariamente por 4). Por lo tanto, el conjunto mostrado es una clase de congruencia (**módulo 5**) [de hecho,  $\mathbb{Z}_{51}$ ] Y NO (módulo 4).

☐ c.  $\{4*k + 3 \mid (\forall k \in \mathbb{Z})\}$ ☐ d.  $\{4*k \mid (\forall k \in \mathbb{Z})\}$ **¡Correcto!**[◀ Video Instructivo 01: Cómo responder un Cuestionario en el Aula Virtual](#)

Ir a...

[Guía de Ejercicios Prácticos - Ficha 02 ▶](#)