

Comenzado el	viernes, 21 de abril de 2023, 16:26
Estado	Finalizado
Finalizado en	viernes, 21 de abril de 2023, 16:54
Tiempo empleado	27 minutos 26 segundos
Puntos	15/19
Calificación	8 de 10 (79%)

Pregunta 1

Correcta

Se puntúa 1 sobre 1



¿Cuáles fueron los aportes que realizaron *George Boole* y *Augustus De Morgan* en el campo matemático del tratamiento de las relaciones lógicas?

Seleccione una:

- ☐ a. *Boole* sentó las bases del Álgebra de Boole, y *De Morgan* demostró que el Álgebra de Boole es válida.
- ☐ b. *De Morgan* sentó las bases del Álgebra de De Morgan, y *Boole* planteó y demostró las leyes de Boole para negar conjunciones y disyunciones.
- ☒ c. *Boole* sentó las bases del Álgebra de Boole, y *De Morgan* planteó y demostró las leyes de De Morgan para negar conjunciones y disyunciones. ✓
- ☐ d. *Boole* sentó las bases de la aritmética en sistema binario, y *De Morgan* usó el sistema binario para diseñar la primera computadora digital de la historia.

Pregunta 2

Correcta

Se puntúa 1 sobre 1

En general, una **expresión** es una fórmula en la cual se usan *operadores* (como suma, resta, comparaciones, etc.) sobre diversas variables y constantes (que reciben el nombre de *operandos* de la expresión). Son ejemplos válidos los siguientes: $3 * a + 2$, $b / c - 4$, $(7 - r) / (4 + a)$, $a > b$, $x + 2 \geq 10$.

¿Es correcta la siguiente definición?

"Una **expresión lógica** es una expresión en la cual el resultado final **es un número**"

Seleccione una:

- ☐ Verdadero
- ☒ Falso ✓

Pregunta **3**

Correcta

Se puntúa 2 sobre 2

¿Cuáles de las siguientes expresiones lógicas son verdadera **si y solo si** el valor de la variable **x** es 4, 5, 6, o 7? (Aclaración: **más de una** puede ser válida, por lo que seleccione **todas** las que considere correctas).

Seleccione una o más de una:

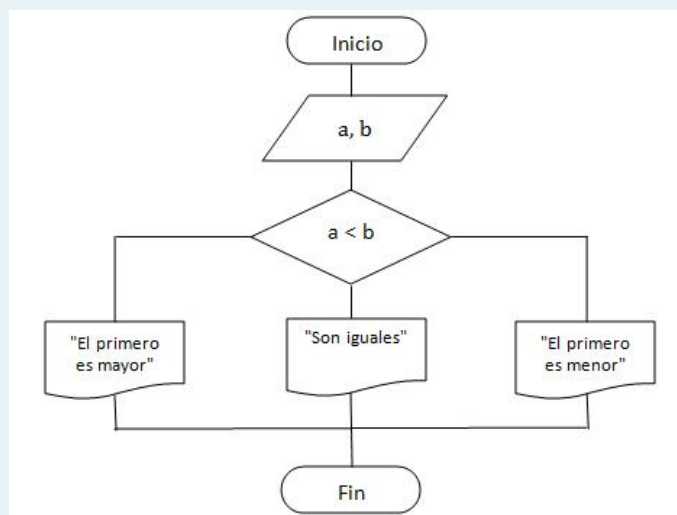
- ☐ a. $x \leq 3$ and $x \geq 7$
- ☐ b. $x == 4$ and $x == 5$ and $x == 6$ and $x == 7$
- ☒ c. $x == 4$ or $x == 5$ or $x == 6$ or $x == 7$ ✓
- ☐ d. $x \neq 1$ and $x \neq 2$ and $x \neq 3$ and $x \neq 4$

Pregunta **4**

Correcta

Se puntúa 1 sobre 1

Suponga que se desea desarrollar un programa que cargue dos números y muestre un mensaje indicando si el primero es menor, igual o mayor al segundo. ¿Está bien planteado el siguiente diagrama de flujo?



Seleccione una:

- ☐ a. Sí. El diagrama está correctamente planteado.
- ☒ b. Está mal planteado: la condición está mal dibujada, ya que una condición no puede tener tres salidas o ramas. ✓
- ☐ c. Está mal planteado: el símbolo usado para la carga de datos, debió ser un rectángulo y no un paralelogramo.
- ☐ d. Está mal planteado: en la condición, se debía preguntar si $a \geq b$.

Pregunta 5

Parcialmente correcta

Se puntúa 1 sobre 2

Sean las siguientes variables:

`a, b, c = 5, 15, -2`

¿Cuáles de las siguientes expresiones lógicas obtendrán un valor final **True** usando las variables y valores aquí indicados? (Observación: puede haber **VARIAS** respuestas correctas... marque **TODAS** las que considere válidas)

Seleccione una o más de una:

- ☐ a. `a == c and b == 15 and c < 8`
- ☐ b. `a >= b or a == b//3 and (c == -2 or a == 14)`
- ☐ c. `a != b and b != 0 and c >= 1`
- ☒ d. `a == 5 or b >= 150 or c != 0` ✓

Pregunta 6

Correcta

Se puntúa 1 sobre 1

¿Por qué motivo debe *indentarse* (*encolumnarse hacia la derecha*) correctamente cada rama de una instrucción condicional en Python?

Seleccione una:

- ☐ a. Para que Python pueda reconocer qué instrucciones pertenecen a cada rama. Pero aún así, la indentación sólo es obligatoria en las ramas que tengan más de una instrucción.
- ☐ b. Para que Python pueda reconocer qué instrucciones pertenecen a cada rama. Pero aún así, la indentación sólo es obligatoria en las ramas que tengan una y sólo una instrucción.
- ☐ c. No es cierto que se deba indentar cada rama. La indentación sólo se sugiere por razones de claridad.
- ☒ d. Para que Python pueda reconocer qué instrucciones pertenecen a cada rama. ✓

Pregunta 7

Incorrecta

Se puntúa 0 sobre 1

Esta pregunta está orientada a la aplicación de las Leyes de Morgan para negar una expresión lógica formada por conjunciones y disyunciones. Sea la siguiente expresión, sumiendo que las variables que se indican están correctamente inicializadas y con valores numéricos:

```
r = not(a < c or b == 5*d + 2 or not(e >= a and f != d) or d < c // 4)
```

¿Cuál de las siguientes expresiones lógicas es equivalente a la expresión anterior? (o sea, ¿cuál de ellas tiene la misma tabla de verdad?) Aplique las Leyes de De Morgan con paciencia y cuidado. Recomendamos asignar valores a las variables, y probar con cada expresión que logre obtener para comparar los resultados que surjan.

Seleccione una:

- ☐ a. `r = a >= c and b != 5*d + 2 and e >= a and f != d and d >= c // 4`
- ☐ b. `r = a >= c or b != 5*d + 2 or e >= a or f != d or d >= c // 4`
- ☒ c. `r = a >= c and b != 5*d + 2 and e < a or f == d and d >= c // 4` ✗
- ☐ d. `r = a >= c and b != 5*d + 2 and e < a and f == d and d >= c // 4`

Pregunta **8**

Correcta

Se puntúa 2 sobre 2

¿Qué tiene de malo el siguiente script en Python?

```
x1 = int(input('Primer valor: '))
x2 = int(input('Segundo valor: '))

if x1 = x2:
    print('Son iguales')
else:
    print('No son iguales')
```

Seleccione una:

- ☐ a. Al ejecutar, la condición sale siempre por falso.
- ☒ b. Error de intérprete: usa el operador = (igual) para comparar, cuando debió usar el == (doble igual). ✓
- ☐ c. Los mensajes que muestra en ambas ramas están al revés.
- ☐ d. Al ejecutar, la condición sale siempre por verdadero.

Pregunta **9**

Correcta

Se puntúa 1 sobre 1

¿Cuál es el efecto del conector **and** ("y lógico") en una condición?

Seleccione una:

- ☒ a. La condición es verdadera si todas las proposiciones son verdaderas. ✓
- ☐ b. La condición es verdadera si todas las proposiciones son falsas.
- ☐ c. La condición es verdadera si alguna de las proposiciones es verdadera.
- ☐ d. La condición es verdadera si una y sólo una de las proposiciones es verdadera.

Pregunta **10**

Correcta

Se puntúa 1 sobre 1

¿Cuál es el efecto del operador **or** ("o lógico") en una condición?

Seleccione una:

- ☒ a. La condición es verdadera si al menos una de las proposiciones es verdadera. ✓
- ☐ b. La condición es verdadera si y sólo si una única proposición es verdadera. Si más de una es verdadera, la salida es False.
- ☐ c. La condición es verdadera si todas las proposiciones son falsas.
- ☐ d. La condición es verdadera si y sólo si todas las proposiciones son verdaderas.

Pregunta **11**

Correcta

Se puntúa 2 sobre 2

El siguiente script en Python, pretende dejar en la variable *may* el mayor de los valores contenidos en las variables *n1* y *n2*. De acuerdo a esto... ¿Hay algún problema con el script mostrado?

```
n1 = int(input('Primer valor: '))
n2 = int(input('Segundo valor: '))

if n1 > n2:
    may = n1
else:
    may = n1

print('Mayor: ', may)
```

Seleccione una:

- ☐ a. No hay ningún problema: calcula y muestra correctamente el mayor, en todos los casos.
- ☒ b. Está mal planteada la rama falsa: está asignando n1 cuando debería asignar n2. ✓
- ☐ c. Está mal escrita la rama falsa: debió usar *elif* en lugar de *else*.
- ☐ d. Está mal planteada la rama verdadera: está asignando n1 cuando debería asignar n2.

Pregunta **12**

Correcta

Se puntúa 2 sobre 2

Suponga que se le pide desarrollar un programa que sea capaz de *elegir aleatoriamente una carta cualquiera de una (y sólo una) baraja española*. ¿Hay algún inconveniente con el programa que les mostramos aquí, o en líneas generales cumple con el requerimiento?

```
__author__ = 'Cátedra de AED'

import random

# Título principal...
print('Selección aleatoria de una carta de la baraja española...')

# Selección del número de la carta...
n = random.randint(1, 12)

# Selección del palo de la carta...
palos = 'Espada', 'Basto', 'Oro', 'Copa'
p = random.choice(palos)

# Visualización de resultados...
print('La carta seleccionada es:')
print('Palo:', p, '- Valor:', n)
```

Seleccione una:

- ☐ a. El programa no cumple correctamente con el requerimiento: a veces selecciona incorrectamente el palo de la carta.
- ☒ b. El programa cumple con el requerimiento (aunque un inconveniente es que podría repetir la misma carta en dos ejecuciones diferentes o repitiendo el mismo esquema en el mismo programa). ✓
- ☐ c. El programa no cumple correctamente con el requerimiento: a veces selecciona incorrectamente el número o valor de la carta.
- ☐ d. El programa cumple con el requerimiento sin ningún inconveniente, seleccionando siempre cartas diferentes en ejecuciones diferentes o repitiendo el mismo esquema en el mismo programa.

Pregunta **13**

Incorrecta

Se puntúa 0 sobre 2

¿Qué hace el script que se muestra en el siguiente esquema?

```
import random

print('Ejemplo de uso de random.random()...')
f = random.random()
i = int(f * 10) + 1
print('El valor generado es:', i)
```

Seleccione una:

- ☐ a. Genera aleatoriamente un número en coma flotante en el intervalo [1, 10], y lo asigna en la variable *i*.
- ☒ b. Genera aleatoriamente un número en coma flotante en el intervalo [0, 1), y lo asigna en la variable *i*. ❌
- ☐ c. Genera aleatoriamente un número entero en el intervalo [1, 10], y lo asigna en la variable *i*.
- ☐ d. Genera aleatoriamente un número entero en el intervalo [0, 9], y lo asigna en la variable *i*.

[◀ Ficha Opcional 04 \[Control de Layout\] \(PDF - para lectura directa\)](#)

Ir a...



[Desafío 01 \[Problema: Conversión de Unidades de Tiempo\] ▶](#)