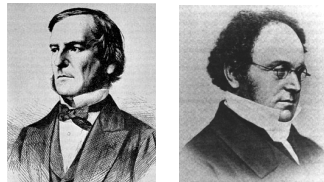


Comenzado el	martes, 24 de abril de 2018, 00:10
Estado	Finalizado
Finalizado en	martes, 24 de abril de 2018, 19:56
Tiempo empleado	19 horas 46 minutos
Puntos	20/20
Calificación	10 de 10 (100%)

Pregunta 1

Correcta

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.
Boole sentó las bases del Álgebra de Boole, y *De Morgan* planteó y demostró las leyes de De Morgan para negar conjunciones y disyunciones. ✓
¡Ok!
- ☐ c.
De Morgan sentó las bases del Álgebra de De Morgan, y *Boole* planteó y demostró las leyes de Boole 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.

¡Correcto!

La respuesta correcta es:

Boole sentó las bases del Álgebra de Boole, y *De Morgan* planteó y demostró las leyes de De Morgan para negar conjunciones y disyunciones.

Pregunta 2

Correcta

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 ✓

¡Correcto!

La respuesta correcta es 'Falso'

Pregunta 3

Correcta

Puntúa 2 sobre 2

¿Cuál de las siguientes expresiones lógicas es verdadera **si y sólo si** el valor de la variable x es 1, 2, 3, o 4?

Seleccione una:

- ☒ a.
 $x == 1 \text{ or } x == 2 \text{ or } x == 3 \text{ or } x == 4$ ✓
- ¡Ok!
- ☐ b.
 $x != 1 \text{ and } x != 2 \text{ and } x != 3 \text{ and } x != 4$
- ☐ c.
 $x == 1 \text{ and } x == 2 \text{ and } x == 3 \text{ and } x == 4$
- ☐ d.
 $x < 0 \text{ and } x > 5$

¡Correcto!

La respuesta correcta es:

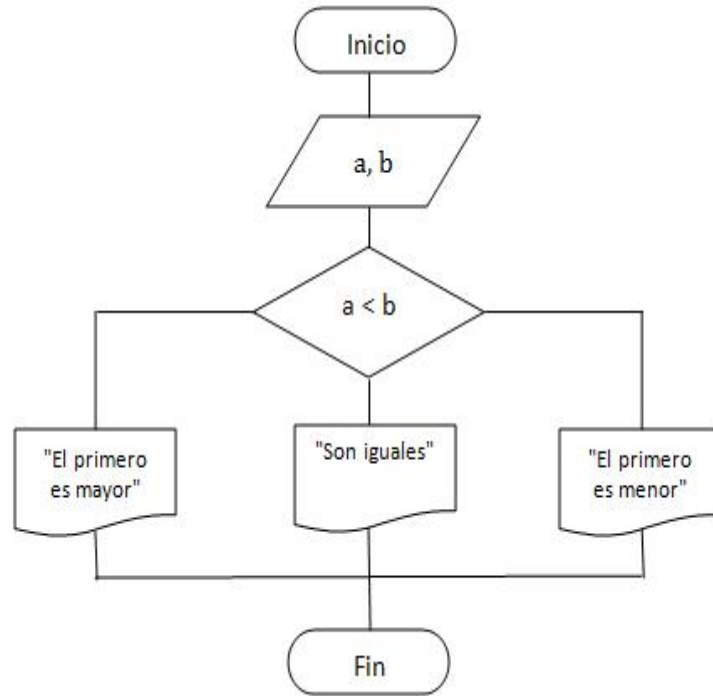
$x == 1 \text{ or } x == 2 \text{ or } x == 3 \text{ or } x == 4$

Pregunta 4

Correcta

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

¡Correcto!

La respuesta correcta es: Está mal planteado: la condición está mal dibujada, ya que una condición no puede tener tres salidas o ramas.

Pregunta 5

Correcta

Puntúa 2 sobre 2

Sean las siguientes variables:

`a, b, c = 3, 10, 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 == 10 and c != 8`

☒ b.

`a == 3 or b > 100 or c != 2` ✓

¡Ok!

☒ c.

`a != b and b != 0 and c >= 1` ✓

¡Ok!

☐ d.

`a >= b or b == 2*c or (c == 2 and a == 4)`

¡Correcto!

Las respuestas correctas son:

`a != b and b != 0 and c >= 1,`

`a == 3 or b > 100 or c != 2`

Pregunta 6

Correcta

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.
No es cierto que se deba indentar cada rama. La indentación sólo se sugiere por razones de claridad.
- ☐ 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.
Para que Python pueda reconocer qué instrucciones pertenecen a cada rama. ✓
¡Ok!
- ☐ d.
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.

¡Correcto!

La respuesta correcta es:

Para que Python pueda reconocer qué instrucciones pertenecen a cada rama.

Pregunta 7

Correcta

Puntúa 1 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 obtenga.

Seleccione una:

☐ a.
 $r = a \geq c \text{ and } b \neq 5*d + 2 \text{ and } e < a \text{ and } f == d \text{ and } d \geq c // 4$

☐ b.
 $r = a \geq c \text{ and } b \neq 5*d + 2 \text{ and } e < a \text{ or } f == d \text{ and } d \geq c // 4$

☒ c.
 $r = a \geq c \text{ and } b \neq 5*d + 2 \text{ and } e \geq a \text{ and } f \neq d \text{ and } d \geq c // 4$ ✓

¡Ok!

☐ d.
 $r = a \geq c \text{ or } b \neq 5*d + 2 \text{ or } e \geq a \text{ or } f \neq d \text{ or } d \geq c // 4$

¡Correcto!

La respuesta correcta es:

```
r = a >= c and b != 5*d + 2 and e >= a and f != d and d >= c // 4
```

Pregunta 8

Correcta

Puntúa 3 sobre 3

¿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.
Al ejecutar, la condición sale siempre por verdadero.
- ☐ c.
Los mensajes que muestra en ambas ramas están al revés.
- ☒ d.
No compila: usa el operador = para comparar, cuando debió usar el == (doble igual). ✓
¡Ok! efectivamente, en Python esto provoca un error de intérprete. A diferencia de otros lenguajes, en Python el operador de asignación no puede usarse en una expresión lógica, incluso si las variables fuesen booleanas.

¡Correcto!

La respuesta correcta es:

No compila: usa el operador = para comparar, cuando debió usar el == (doble igual).


Pregunta 9

Correcta

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.

¡Ok!
- ☐ b.
La condición es verdadera si alguna de las proposiciones es verdadera.
- ☐ c.
La condición es verdadera si todas las proposiciones son falsas.
- ☐ d.
La condición es verdadera si una y sólo una de las proposiciones es verdadera.

¡Correcto!

La respuesta correcta es:

La condición es verdadera si todas las proposiciones son verdaderas.

Pregunta 10

Correcta

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

¡Correcto!

La respuesta correcta es:

La condición es verdadera si al menos una de las proposiciones es verdadera.

Pregunta 11

Correcta

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

¡Correcto!

La respuesta correcta es:

Está mal planteada la rama falsa: está asignando n1 cuando debería asignar n2.

Pregunta 12

Correcta

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 cumple con el requerimiento sin ningún inconveniente, seleccionando siempre cartas diferentes en ejecuciones diferentes o repitiendo el mismo esquema en el mismo programa.
- ☒ 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). ✓
¡Ok! Efectivamente, selecciona bien una carta cualquiera, pero si se ejecuta nuevamente (o se repite el mismo esquema en el mismo programa) todas las cartas vuelven a estar disponibles y podría volver a seleccionar la misma.
- ☐ c.
El programa no cumple correctamente con el requerimiento: a veces selecciona incorrectamente el número o valor de la carta.
- ☐ d.
El programa no cumple correctamente con el requerimiento: a veces selecciona incorrectamente el palo de la carta.

¡Correcto!

La respuesta correcta es:

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).

Pregunta 13

Correcta

Puntúa 2 sobre 2

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

```
__author__ = 'Cátedra de AED'
```

```
import random
```

```
print('Ejemplo de uso de random.random()...')
```

```
f = random.random()
```

```
i = int(f * 10)
```

```
print('El valor generado es:', i)
```

Seleccione una:

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

¡Ok! Efectivamente... lo primero que hace es invocar a `random.random()` y obtiene con eso un número con decimales entre 0 y 1... pero luego lo multiplica por 10 (con lo cual la coma se corre a la derecha exactamente un lugar, eliminando el cero y tomando el primer decimal para la parte entera) y finalmente trunca los decimales con la función `int()`. El resultado es un número entero de un sólo dígito, en [0, 9].
- ☐ d.
Genera aleatoriamente un número entero en el intervalo [1, 10], y lo asigna en la variable *i*.

¡Correcto!

La respuesta correcta es:

Genera aleatoriamente un número entero en el intervalo [0, 9], y lo asigna en la variable *i*.

