 UNIVERSIDAD GALILEO DE GUATEMALA

MATEMÁTICA PARA DESARROLLADORES

# Tarea #1 Resuelta

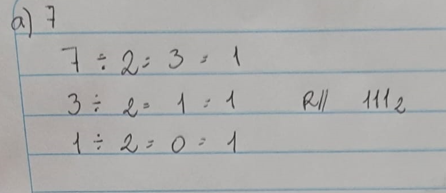
Lógica matemátoca y proposicional

**Descripción:**

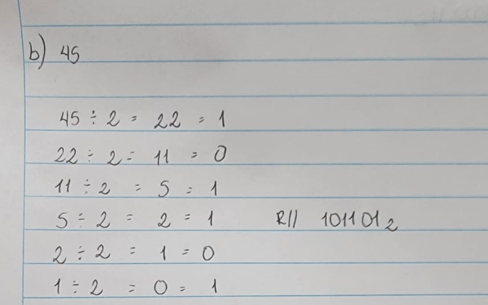
En la realización de esta prácAca debe desarrollar los enunciados presentados a conAnuación, debe uAlizar los procedimientos de la forma más ordenada y explícita posible.

**Primera sección**

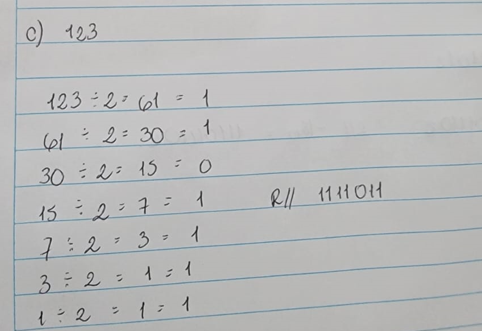
1. UAlice el procedimiento visto en clase para la conversión de los siguientes números decimales a binarios:
   1. 7



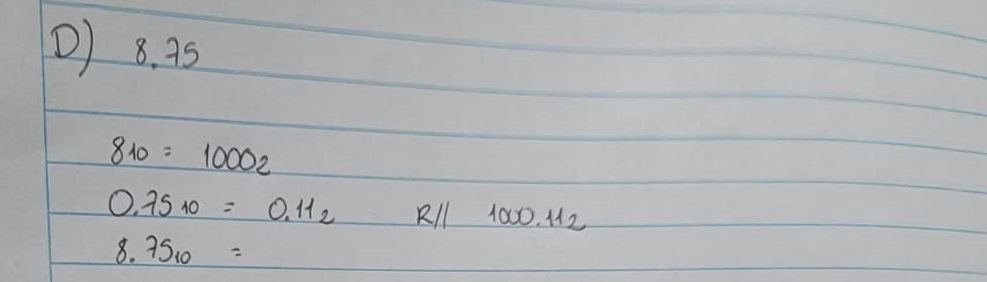
* 1. 45



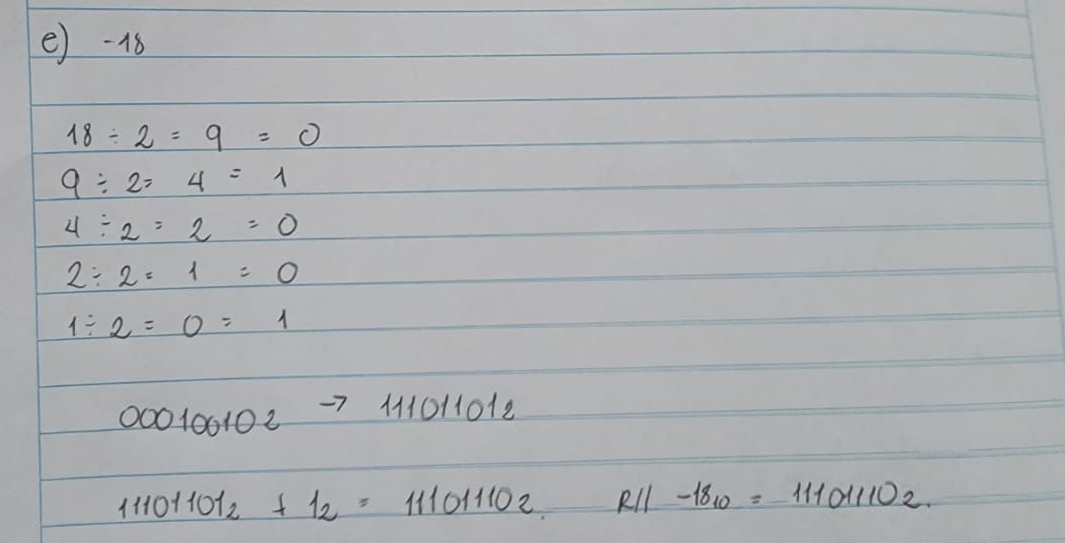
* 1. 123



* 1. 8.75



* 1. -18 (uAlice complemento a dos, ver un ejemplo en la sección de anexos de este documento).



1. Realice las tablas de verdad de las siguientes operaciones de lógica booleana, ver un ejemplo en anexos:
   1. A ^ B

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A ^ B** |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

* 1. (¬A ^ B) v C

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **¬A** | **¬A ^ B** | **(¬A ^ B) v C** |
| T | T | T | F | F | T |
| T | T | F | F | F | F |
| T | F | T | F | F | T |
| T | F | F | F | F | F |
| F | T | T | T | T | T |
| F | T | F | T | T | T |
| F | F | T | T | F | T |
| F | F | F | T | F | F |

* 1. ¬(A v C) ^B

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **A v C** | **¬(A v C)** | **¬(A v C) ^ B** |
| T | T | T | T | F | F |
| T | T | F | T | F | F |
| T | F | T | T | F | F |
| T | F | F | T | F | F |
| F | T | T | T | F | F |
| F | T | F | F | T | T |
| F | F | T | T | F | F |
| F | F | F | F | T | F |

* 1. (B ^ C)^(¬A)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **A v C** | **¬(A v C)** | **¬(A v C) ^ B** |
| T | T | T | T | F | F |
| T | T | F | T | F | F |
| T | F | T | T | F | F |
| T | F | F | T | F | F |
| F | T | T | T | F | F |
| F | T | F | F | T | T |
| F | F | T | T | F | F |
| F | F | F | F | T | F |

* 1. (M ^ (¬A)) v (¬C)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **B ^ C** | **¬A** | **(B ^ C) ^ (¬A)** |
| T | T | T | T | F | F |
| T | T | F | F | F | F |
| T | F | T | F | F | F |
| T | F | F | F | F | F |
| F | T | T | T | T | T |
| F | T | F | F | T | F |
| F | F | T | F | T | F |
| F | F | F | F | T | F |

**Segunda sección**

A conAnuación, se presentan ejercicios los cuales deberá resolver uAlizando el lenguaje JavaScript uAlizando NodeJS.

Ejercicio 1: Sistema de Evaluación de Riesgo

Cree una función que evalúe el nivel de riesgo de un cliente bancario basado en varios factores.

* Defina la función *evaluateRisk(income, hasLoans, latePayments, isStudent)*:

o Un cliente es considerado de alto riesgo si:

* + - * + Tiene ingresos menores a $20,000 y más de 3 pagos atrasados.
        + O Aene préstamos acAvos y es estudiante.

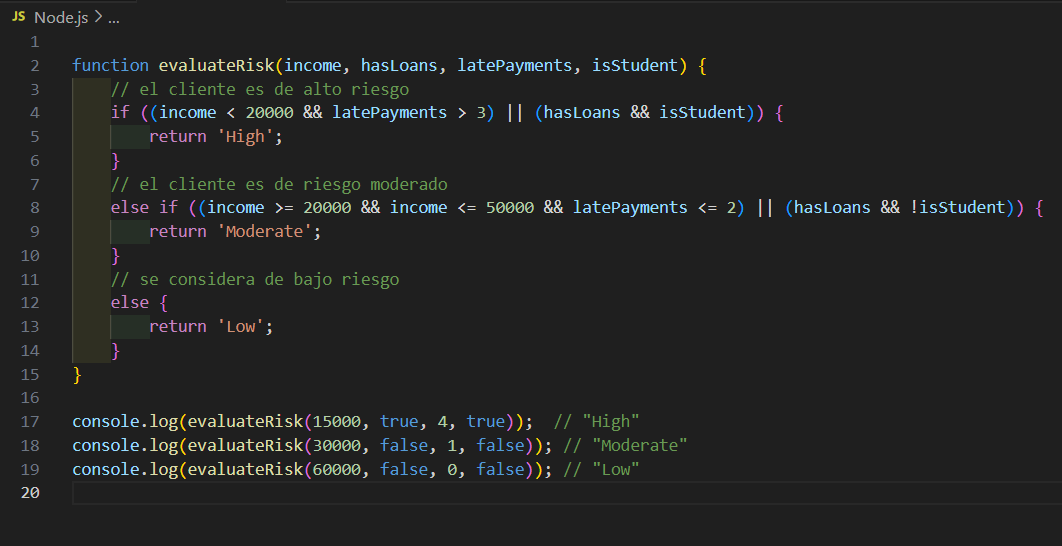
o Un cliente es considerado de riesgo moderado si:

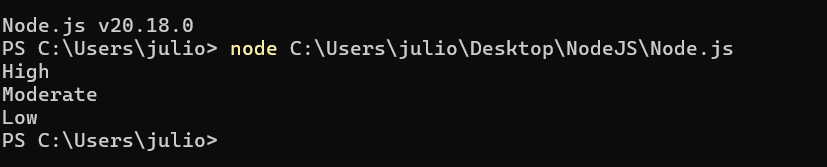
* + - * + Tiene ingresos entre $20,000 y $50,000, y no Aene más de 2 pagos atrasados.
        + O si Aene préstamos pero no es estudiante.

o Si no cumple ninguna de estas condiciones, se considera de bajo riesgo.

* Retorne el nivel de riesgo como un string: 'High', 'Moderate' o 'Low'.
* Por ejemplo, el uso de la función sería:

o console.log(evaluateRisk(15000, true, 4, true)); // "High"





Ejercicio 2: Motor de Recomendación de Productos

Diseñe una función para recomendar productos a un usuario basado en varios factores.

* Cree una función *recommendProduct(age, isMember, purchaseHistory)*:

o Un producto de alta tecnología se recomienda si:

* + - * + La persona es miembro y ha comprado al menos 5 productos tecnológicos antes.
        + O si la persona Aene entre 18 y 30 años y ha comprado 2 o más productos de moda.

o Un producto de moda se recomienda si:

* + - * + La persona no es miembro, pero ha comprado al menos 3 productos antes.
        + O si Aene entre 25 y 40 años.
        + En cualquier otro caso, recomienda un producto genérico.
* purchaseHistory es un objeto con la canAdad de productos comprados por categoría: { tech: number, fashion: number, other: number }.
* Por ejemplo, el uso de la función sería:

o console.log(recommendProduct(22, true, { tech: 6, fashion: 1, other: 2 })); // "High-Tech Product"

**Rúbrica de calificación**

Para la entrega deberá incluir únicamente el link de un repositorio con el procedimiento debidamente explicado, todo el procedimiento escrito debe ser agregado en el “README.md” del proyecto. Para la primera sección puede insertar imágenes si su procedimiento es a mano o escribirlo digitalmente. Para la segunda sección, deberá incluir todo el código debidamente funcional y explicado. Si no cumple con alguna de estas condiciones la tarea será anulada.

|  |  |
| --- | --- |
| **Descripción** | **Puntaje** |
| Primera sección, ejercicio 1 | 25% |
| Primera sección, ejercicio 2 | 25% |
| Segunda sección, ejercicio 1 | 25% |
| Segunda sección, ejercicio 2 | 25% |
| **Total** | 100% |

**Entrega de ac9vidad:**

La acAvidad podrá ser entregada únicamente como fecha máxima el día jueves **31/10/2024 a medianoche**. Si no se presenta este día la acAvidad ya no será calificada.

**Anexos**

* **Procedimiento para complemento a dos**

Para este ejemplo de hará la conversión del número -5.

* 1. ConverAr el número posiAvo a binario, dividiendo el número posiAvo dentro de dos y sus resultados enteros correspondientes (puede uAlizar el método de la escalera visto en clase):

5⁄2 = 2 ; // residuo: 1 2⁄2 = 1; // residuo: 0

1⁄2 = 0; // residuo: 1

Por lo tanto 5 (decimal) = 0000 0101 (bintario, 8 bits)

* 1. Obtener el complemento a uno, este se obAene cambiando todos los 1 por 0 y todos los 0 por 1:

Complemento a uno de 0000 0101 = 1111 1010

* 1. Sumar 1 para obtener el complemento a dos

1111 1010 + 0000 0001 = 1111 1011

* 1. El resultado final será:

-5 (decimal) = 1111 1011 (binario de 8 bits)

* **Construcción de una tabla de verdad**

Por ejemplo, tome la expresión booleana: (AvC)^B

* 1. Para resolverla, definiendo cada componente y evaluar su valor de verdad:
     + 1. A: proposición
       2. B: proposición
       3. C: proposición
       4. A v B: La disyunción de A y B (AND)
       5. (A v B) ^C: La conjunción de la disyunción con C.

* 1. Construir la tabla de verdad

Dado que se Aenen tres proposiciones (A, B y C), hay 8 combinaciones posibles de valores de verdad para A, B y C. La tabla de verdad se vería de la siguiente forma:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | | **B** | | **C** | | **A v B** | | **(A v B) ^ C** | |
| T | | T | | T | | T | | T | |
| T | | T | | F | | T | | F | |
| T | | F | | T | | T | | T | |
| T | | F | | F | | T | | F | |
| F | | T | | T | | T | | T | |
| F | | T | | F | | T | | F | |
| F | | F | | T | | F | | F | |
| F | | F | | F | | F | | F | |

* 1. Evaluación de la tabla de verdad:

La columna A v B: La disyunción es verdadera (T) si al menos uno de A o B es verdadero. La disyunción es falsa (F) solo si ambos, A y B, son falsos.

La columna (A v B) ^C es verdadera (T) solo cuando A v B como C son verdaderos y es false (F) si cualquiera de A v B o C es falso.