

ASIGNATURA: LENGUAJES DE PROGRAMACIÓN
PERÍODO ACADÉMICO: 2023-1
FECHA: 05/05/2023
TIEMPO: 100 minutos

| NOTA |
|------|
| |

LABORATORIO Nº 1

| CÓDIGO | APELLIDOS Y NOMBRES | SECCIÓN |
|--------|---------------------|---------|
| | | |

ANTES DE INICIAR LA EVALUACIÓN DEBE LEER LAS INSTRUCCIONES

Si la evaluación indica cargar algún archivo en la computadora, recuerde que es responsabilidad del estudiante hacerlo en el tiempo establecido y con las instrucciones dadas. Debe indicar el número de la misma en el recuadro siguiente:

INSTRUCCIONES GENERALES:

- El examen consta de 6 preguntas, el puntaje se indica en cada una de ellas.
- Genere un documento solo con sus respuestas y súbalo utilizando Blackboard.
- Debe utilizar los métodos y la forma de programación utilizada durante las clases del curso.
- El procedimiento, el orden, la claridad de las respuestas y el uso apropiado del lenguaje (notaciones, símbolos y unidades), serán considerados como criterios de calificación.
- **Leer detenidamente las situaciones que ocasionarán la anulación de la evaluación, que se encuentran a continuación.**

SITUACIONES QUE OCASIONARÁN LA ANULACIÓN DE LA EVALUACIÓN:

- Utilizar funciones y/o forma de programar diferente a lo realizado en clase.
- Consultar material no autorizado o hacer uso de Internet.
- Abrir cualquier aplicativo no autorizado. Solo se puede utilizar VSCode y su editor de texto.
- Entregar respuestas generadas por inteligencia artificial.

Los profesores de la asignatura

FIRMA DEL ALUMNO
(LEYÓ LAS INSTRUCCIONES)

Verificación de integridad en transmisiones de datos

En cualquier sistema de comunicación, es muy importante que los datos puedan llegar a su destino de manera precisa y sin errores. Sin embargo, distintos factores como el ruido eléctrico, la interferencia electromagnética, entre otros, puede producir que algunos datos se pierdan o puedan corromperse. En contextos específicos como por ejemplo el sector financiero, un error en estos datos transmitidos puede tener un impacto muy grande.

Una de las formas para poder realizar la verificación de integridad de los datos transmitidos es a través de un algoritmo, el cual se basa en el conteo de la cantidad de bits 1 en una secuencia de datos. Deben tenerse las siguientes consideraciones:

- A. Los datos deberán estar en formato binario y se agregará un bit adicional en el lado del emisor (quien envía los datos) según lo especificado en la parte B.
- B. Contar la cantidad de bits 1 presentes en la secuencia de datos. Si la cantidad es impar, agregar un bit adicional con el valor 1. En caso contrario, agregar un bit con el valor 0.

Por ejemplo, para el valor 7:

- Representación en binario 111.
- Como hay un número impar de bits 1, se agrega un bit adicional con el valor 1.
- El dato completo para transmitir sería 1111.

- C. En el lado del receptor se realizará la comprobación contando la cantidad de bits 1 en los datos recibidos (sin incluir el bit agregado). Análogamente, se calculará un bit adicional. Si la cantidad de bits 1 es impar, se considerará un valor de 1 y en caso contrario un valor de cero. Se comparará el valor obtenido con el último bit transmitido (el que se agregó por parte del emisor). Si coinciden, los datos habrán sido transmitidos correctamente.

Por ejemplo, si asumimos que la secuencia llega correctamente, habremos leído 1111. Separando los datos, obtendremos 111 como dato de entrada y 1 como bit adicional.

La suma $(1+1+1)$ nos dará una cantidad impar de bits, por lo que se considera un valor 1 como bit adicional.

Dicho valor 1, coincide con el último bit agregado por el emisor (ver sección B). Por lo tanto, podemos decir que, según este algoritmo, los datos fueron transmitidos correctamente.

Teniendo en cuenta este contexto se le solicita responder las siguientes preguntas:

PREGUNTA 1 (3 puntos)

Para poder preparar la transmisión de datos, es necesario convertir los valores a su representación binaria. En ese contexto se le solicita implementar la función **convertirBinario** que reciba un número entero y retorne su representación binaria. Por simplicidad se le solicita retornar la representación en forma de una lista de números enteros.

Ejemplo:

Para $n = 30$, su representación como binario es 11110.

Por lo tanto, la respuesta será [1, 1, 1, 1, 0]

PREGUNTA 2 (3 puntos)

Implemente la función **agregarBitAdicional** que implementa el paso B) del enunciado. Tome en cuenta que el parámetro de entrada será una lista de bits (del mismo formato que en la pregunta anterior). Su respuesta deberá ser la misma lista, adicionándole el bit según las reglas especificadas. Debe utilizar recursión de cola.

Ejemplo:

Para la representación anterior: [1, 1, 1, 1, 0] la cantidad de dígitos 1 es par, por lo tanto deberemos agregar un bit 0 al final.

Con ello, la respuesta sería [1, 1, 1, 1, 0, 0]

PREGUNTA 3 (3 puntos)

Implemente la solución de la pregunta 1) utilizando recursión de cola. Comente la razón por la cual la solución brindada utiliza recursión de cola. Sea preciso en el uso de definiciones.

PREGUNTA 4 (4 puntos)

Implemente la función **validarTransmision** que implementa el paso C) del enunciado. Tome en cuenta que el parámetro de entrada será una lista de bits (en el mismo formato de las preguntas anteriores). Su función debe retornar verdadero o falso dependiendo de si la transmisión se realizó de forma correcta o no.

Ejemplo:

Para el mismo caso anterior, considerando lista = [1, 1, 1, 1, 0, 0].

Se observa que los datos transmitidos son [1, 1, 1, 1, 0] y 0 es el bit de validación. Sumando los bits encontramos que $1 + 1 + 1 + 1 + 0 = 4$. Siendo una cantidad par, el nuevo bit calculado es cero, el cual coincide con el bit de validación leído (último elemento de la lista).

PREGUNTA 5 (4 puntos)

Proponga una nueva versión de la función implementada en la pregunta 2, denomínela **agregarBitAdicionalCuatro**. En esta versión se agregarán 2 bits adicionales, los que corresponderán al módulo de la división entre 4 de la cantidad de bits 1 contados.

Ejemplo:

Para el mismo caso tratado [1, 1, 1, 1, 0], la cantidad de bits 1 es 4. Por lo tanto, el resultado de la operación módulo ($4 \bmod 4$) es cero. Se agrega este valor cero con 2 bits, es decir, como [0, 0]. El resultado final será [1, 1, 1, 1, 0, 0, 0].

Observe que en todos los casos podrá registrar los residuos posibles como 2 bits, para 1 sería [0, 1], para 2 sería [1, 0] para 3 y [1, 1] para 4.

PREGUNTA 6 (3 puntos)

Implemente la función **validarTransmisionCuatro** que implementa la validación haciendo uso de los 2 bits de validación generados en la pregunta anterior.

Ejemplo:

Para el caso anterior: [1, 1, 1, 1, 0, 0, 0] tenemos que los datos originales son [1, 1, 1, 1, 0] y el residuo es cero (los 2 últimos valores). La suma de bits 1 es cuatro, por lo que coincide con el valor cero calculado. En este contexto, la función debería retornar verdadero.