

Examen – 21 de diciembre de 2023

- **Duración del examen: 3 Hs.**
- **No** se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc). **Apague** su teléfono celular.
- **Escriba las hojas de ambos lados con buena letra.** Las partes no legibles se considerarán no escritas.
- Sólo se contestarán preguntas sobre interpretación de la letra.
- **En la primera hoja, en el ángulo superior derecho, ponga su nombre, número de cédula y cantidad de hojas. En las demás hojas ponga nombre y número de cédula. En todas las páginas ponga número de página.**

Para la resolución de los diferentes ejercicios **solamente** podrá utilizar las siguientes funciones brindadas por **Octave**:

- `length()` y `size()`
- `mod()` y `rem()`
- `fix()`, `floor()`, `ceil()` y `round()`
- `zeros()`, `ones()`, `isempty()`

Notas: - En todos los ejercicios se deben usar las estructuras de control adecuadas para cada caso.
 Por ejemplo: se controlará el uso correcto de `for` y `while`.
 - No se deben realizar más iteraciones o invocaciones recursivas que las necesarias.

Problema 1	12 (1,2,2,3,4) ptos	
-------------------	---------------------	--

Calcule y **justifique** cuando **NO** sea posible, la representación del número -102_{10} en:

- Binario sin signo** utilizando 8 bits en total.
- Complemento a 1** con 8 bits.
- Complemento a 2** con 8 bits.
- Desplazamiento a M** con 8 bits (con $M = 2^{(n-1)} - 1 = 127$).
- Punto flotante** con 1 bit de signo, 3 bits de exponente y 6 bits de mantisa (exponente con $M = 3$).

Problema 2	6 ptos	
-------------------	--------	--

Suma de matrices dispersas:

¿Cuál es la matriz dispersa **C** resultante de sumar las matrices dispersas **A** y **B**?

$A_{fil} = [4 \ 9 \ 5 \ 8]$, $A_{col} = [10 \ 2 \ 6 \ 3]$, $A_{dat} = [\ 1 \ 3 \ 4 \ 9]$
 $B_{fil} = [4 \ 9 \ 7 \ 5]$, $B_{col} = [1 \ 2 \ 5 \ 6]$, $B_{dat} = [-1 \ 3 \ 9 \ -4]$
 $C_{fil} = ?$, $C_{col} = ?$, $C_{dat} = ?$

Problema 3	14 ptos	
-------------------	---------	--

Dada una matriz dispersa **M** en formato elemental **Mf**, **Mc**, **Md**, el número de fila **f** y el vector **FilVec** realice la función **ITERATIVA agregofit** que devuelve la matriz dispersa **R** que es igual a **M** pero con el vector **FilVec** agregado en la fila **f**. La actual fila **f** de **M** y todas las filas siguientes aumentan su valor en 1. **FilVec** es un vector común y por lo tanto puede contener ceros. Ni **M** ni **R** están ordenadas.

agregofit(Mf, Mc, Md, f, FilVec) -> Rf Rc Rd
`agregofit([], [], [], 2, [8 0 9]) -> [2 2] [1 3] [8 9]`
`agregofit([1 2], [2 1], [1 4], 2, [8 0 9]) -> [1 3 2 2] [2 1 1 3] [1 4 8 9]`

Problema 4	22 (10,12) ptos	
-------------------	-----------------	--

- a) Escriba la función **RECURSIVA juntarec** que dado dos vectores **a** y **b** ordenados de menor a mayor, pone sus elementos en un nuevo vector ordenado **res**. Los 3 vectores pueden tener elementos repetidos.

juntarec(a , b) -> res , **juntarec([], []) -> []**
juntarec([1 3 5], []) -> [1 3 5] , **juntarec([1 3 7], [2 3]) -> [1 2 3 3 7]**

- b) Elabore la función **RECURSIVA MSrec** que dado un vector **v** con números, lo ordena de menor a mayor siguiendo el algoritmo llamado "Merge Sort". Este algoritmo divide a **v** en dos vectores de igual largo (si **v** tiene largo impar, tendrán 1 elemento de diferencia), ordena cada vector por separado, y luego los junta en un solo vector ordenado **res**.

Puede usar **juntait** aunque no lo haya realizado en la parte a).

MSrec(v) -> res , **MSrec([]) -> []**
MSrec([1 2 3 2 1]) -> [1 1 2 2 3] , **MSrec([3 2 2 1]) -> [1 2 2 3]**

Problema 5	24 (12,12) ptos	
-------------------	-----------------	--

- a) Escriba una función **ITERATIVA Snula** que dada una matriz densa **M**, devuelve **1** si en todas las filas de **M** la suma de sus elementos vale **0** y devuelve **0** si hay alguna fila de **M** que no suma **0**. La matriz **M** tiene al menos **1** elemento.

Snula(M) -> res , **Snula(0) -> 1** , **Snula(1) -> 0**
Snula([1 -1]) -> 1 , **Snula([1 -1; 0 1]) -> 0** , **Snula([0 -1 1;-2 1 1]) -> 1**

- b) Desarrolle la función **ITERATIVA ORTGit** que dada una matriz **M** y un vector **v**, devuelve la cantidad de filas de **M** que son ortogonales⁽¹⁾ al vector **v**. Se asume que **v** y las filas de **M** tienen igual largo.

ORTGit(M , v) -> res , **ORTGit([1 2], [-2 1]) -> 1**
ORTGit([1 0] , [-2 1]) -> 0 , **ORTGit([1 2; 2 4; -4 8], [-2 1]) -> 2**
ORTGit([1 2; 2 3] , [-2 1]) -> 1 , **ORTGit([1 0; 2 3], [-2 1]) -> 0**
ORTGit([], []) -> 0 , **ORTGit([0 0], [-2 1]) -> 1**

Problema 6	22 (10,12) ptos	
-------------------	-----------------	--

- a) Realice una función **RECURSIVA suma1** que sume **1** a un número binario **b** entero y positivo con al menos un bit. El **0** se puede representar como **[0]** o como **[]**.

suma1(b) -> res , **suma1([]) -> [1]** , **suma1([0]) -> [1]**
suma1([1]) -> [1 0] , **suma1([1 0 1]) -> [1 1 0]** , **suma1([1 1]) -> [1 0 0]**

- b) Implemente la función **RECURSIVA sumaR** que suma dos números binarios **a** y **b** enteros y positivos que pueden tener distinto largo. El **0** se puede representar como **[0]** o como **[]**. Puede usar **suma1** aunque no haya realizado la parte a).

sumaR(a,b) -> res , **sumaR([1 0 1], [1 0]) -> [1 1 1]**
sumaR([1 0], [1 1]) -> [1 0 1] , **sumaR([], []) -> [1]**

¹Dos vectores **a** y **b** de igual largo son ortogonales entre si cuando $\sum_{i=1:L_a} a(i) \cdot b(i) = 0$