

Examen – 20 de diciembre de 2022

- **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()` y M' (la transpuesta)

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 para resolver el problema

Problema 1	10 (1,2,2,2,3) ptos	
-------------------	---------------------	--

Calcule, y justifique cuando no sea posible, la representación del número -101_{10} en:

- binario con signo y valor absoluto utilizando 8 bits en total.
- complemento a 1 con 7 bits.
- complemento a 2 con 7 bits.
- Corrimiento a M con 7 bits (con $M = 2^{n-1}$).
- Punto flotante con 1 bit de signo, 4 bits de exponente y 6 bits de mantisa (exponente con $M=7$)

Problema 2	10 ptos	
-------------------	---------	--

Escriba una función *iterativa* `esMayor(a, b)` que dados dos enteros no negativos en notación **decimal** y representados como vectores (un dígito por posición), que pueden ser de distinto largo, devuelva 1 si $a > b$ y 0 en caso contrario. Ambos vectores tienen al menos un elemento.

Sugerencia: iguale la longitud de ambos vectores agregando ceros.

```
Esmayor( [2], [1]) -> 1 ; Esmayor([0 0 5], [5]) -> 0
Esmayor( [2], [0 1]) -> 1 ; Esmayor( [5], [0 5]) -> 0
Esmayor([1 5], [3 0 1]) -> 0 ; Esmayor([0 0 6], [5]) -> 1
```

Problema 3	22 (9,13) ptos	
-------------------	----------------	--

a) Escriba una función *recursiva* `Divisores(n, v)` que, dado un número n y un vector v de números enteros positivos, devuelva aquellos valores $v(i)$ que son divisores de n (o sea, que al realizar la división entera entre n y $v(i)$ el resto es 0).

```
Divisores(8, []) -> [] ; Divisores(8, [4 4 4 8]) -> [4 4 4 8]
Divisores(8, [1 3 4 7 16]) -> [1 4] ; Divisores(9, [5 7 2]) -> []
```

b) Escriba una función *recursiva* `halloPrimos(n)` que dado un entero $n \geq 2$, devuelva un vector ordenado de menor a mayor, con todos los primos menores o iguales a n .

Sugerencia: Utilice la función `Divisores(n, v)`, aunque no la haya implementado.

Ayuda: Si n es primo, entonces `Divisores(n, halloPrimos(n-1))` devuelve `[]`.

```
halloPrimos(2) -> [2] ; halloPrimos(5) -> [2 3 5]
halloPrimos(3) -> [2 3] ; halloPrimos(13) -> [2 3 5 7 11 13]
halloPrimos(4) -> [2 3] ; halloPrimos(16) -> [2 3 5 7 11 13]
```

Problema 4 20 (10,10) ptos.

a) Escriba una función *iterativa* SumFils(Mv,Mf,Mc,m) que, dada una matriz M dispersa en formato elemental y representada de forma **ordenada**^(*), y un número m, devuelva un número con la suma de todos los elementos de las primeras m filas de M.

```
SumFils([5 3 3],[4 4 8],[3 5 1],5)      -> 8
SumFils([9 1 7 4],[1 1 3 5],[2,5,2,1],4) -> 17
SumFils([],[],[],2)                    -> 0
```

b) Escriba una función *iterativa* SumCols(Mv,Mf,Mc,n) que, dada una matriz M dispersa en formato elemental y representada de forma **ordenada**^(*), y un número n, devuelva un número con la suma de todos los elementos de las primeras n columnas de M.

```
SumCols([7 3 3],[4 4 8],[3 5 1],1)      -> 3
SumCols([5 2 3 4],[1 1 3 4],[2 5 2 1],4) -> 12
SumCols([],[],[],2)                    -> 0
```

(*) Mv = [5 3 7 2 1 1 1] Una matriz cualquiera se representa de forma ordenada cuando:
Mf = [1 2 4 4 4 6 9] - los valores de las filas se ordenan de menor a mayor,
Mc = [7 3 5 7 9 2 8] - Si hay filas iguales, entonces se ordenan de menor a mayor las columnas correspondientes.

Problema 5 15 ptos

Escriba en Octave una función *recursiva* restoUnoR(b) que dado un binario entero b>0 representado como un vector de 0s y 1s, genera el binario inmediatamente inferior. Tanto b como el resultado de la resta tienen al menos un elemento.

Sugerencia: Restar 1 a un binario positivo implica intercambiar 0s por 1s desde el bit menos significativo hasta el primer 1 inclusive.

```
restoUnoR([1 1 0 0]) -> [1 0 1 1] ; restoUnoR([0 1 1]) -> [0 1 0]
restoUnoR([1 0 1]) -> [1 0 0] ; restoUnoR([1 0]) -> [0 1]
restoUnoR([1 0 0]) -> [0 1 1] ; restoUnoR([0 1]) -> [0 0]
```

Problema 6 23 (6,9,8) ptos

a) Escriba una función *recursiva* prefR(v,p), que dado un vector v y otro p, devuelva 1 si el vector v comienza con el vector p (o sea si p es prefijo de v), y 0 en caso contrario.

```
prefR([],[]) -> 1 ; prefR([2 3 4],[3 2]) -> 0
prefR([2 3],[1]) -> 1 ; prefR([2 3 4],[3 4]) -> 0
prefR([2 3 4],[2 3]) -> 1 ; prefR([2 3 4],[2 3 4]) -> 1
prefR([2 3 4],[2 3 4 5]) -> 0
```

b) Escriba una función *recursiva* contR(v,c) que dado un vector v y otro c, devuelva 1 si c está contenido en v, y 0 en caso contrario.

Sugerencia: utilice prefR aunque no lo haya implementado.

```
contR([],[]) -> 1 ; contR([2 3 4],[3 4]) -> 1 ; contR([2 3 4],[2 3 4]) -> 1
contR([2 3 4],[1]) -> 1 ; contR([2 3 4],[2 4]) -> 0 ; contR([2 3 4],[2 3 4 5]) -> 0
```

c) Escriba una función *iterativa* contMit(M,c) que dada una matriz M y un vector c, devuelva 1 si c está contenido en alguna fila o columna de M, y 0 en caso contrario.

Sugerencia: Para simplificar el código, utilice contR aunque no lo haya implementado.

```
M=[ 1 2 3 | contMit(M,[]) -> 1 ; contMit(M,[5 8]) -> 1
  4 5 6 | contMit(M,[4 5]) -> 1 ; contMit(M,[3 6 9]) -> 1
  7 8 9 | contMit(M,[7 9]) -> 0 ; contMit(M,[7 8 9 10]) -> 0
  5 2 8 | contMit(M,[3 6 9 8]) -> 1
```