

## Examen – 15 de febrero de 2024

- **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.

<b>Problema 1</b>	12 (1,1,1,1,1,1,3,3) pts	
-------------------	--------------------------	--

**Justifique sus respuestas:**

- ¿Qué número representa la tira 00000000 en **Complemento a 1** con 8 bits?
- ¿Qué número representa la tira 11111111 en **Complemento a 1** con 8 bits?
- ¿Qué número representa la tira 00000000 en **Complemento a 2** con 8 bits?
- ¿Qué número representa la tira 11111111 en **Complemento a 2** con 8 bits?
- ¿Qué número representa la tira 00000000 en **Desplazamiento a M** con 8 bits (con  $M = 2^7 - 1$ )?
- ¿Qué número representa la tira 11111111 en **Desplazamiento a M** con 8 bits (con  $M = 2^7 - 1$ )?
- Represente el número -0,875 utilizando un sistema de **punto flotante** con 1 bit de signo, 4 bits de exponente (con exceso  $M=7$ ) y 7 bits de mantisa.
- ¿Qué número representa la tira 010101100000 en un sistema de **punto flotante** con 1 bit de signo, 4 bits de exponente (con exceso  $M=7$ ) y 7 bits de mantisa?

<b>Problema 2</b>	6 pts	
-------------------	-------	--

De acuerdo con lo visto en el curso determine el valor de las variables `a`, `b` y `c` luego de ejecutar `miscript.m` desde la línea de comandos de octave.

<pre>% funcAux.m function b = funcAux(a)     if a&lt;5         a = a+1;         b = a;     else         b = a + funcAux(a-1);         a = a-1;     end end</pre>	<pre>% miscript.m b = 1; a = 5; for c = 1:3     b = funcAux(b)+1;     a = a - 1; endfor</pre>
--	---

<b>Problema 3</b>	34 (10,12,12)ptos
-------------------	-------------------

Parte a)

Dado un vector  $\mathbf{v}$ , que contiene ceros y unos, implemente en Octave una función **ITERATIVA primeraSecuencia** que devuelve dos resultados: el número que está en la posición uno del vector y la cantidad de veces que aparece ese número al principio del vector, de forma consecutiva. Asuma que el vector no es vacío.

```

primeraSecuencia(v) -> nro cant

primeraSecuencia([0 1 1 1]) -> 0 1
primeraSecuencia([0 0 1 1 1]) -> 0 2
primeraSecuencia([0 0 1 0 1 1]) -> 0 2
primeraSecuencia([1 1 1]) -> 1 3
primeraSecuencia([1 1 1 0 0]) -> 1 3

```

Parte b)

Dado un vector  $\mathbf{v}$ , que contiene ceros y unos, implemente en Octave una función **ITERATIVA codigo** que devuelve un vector que codifica la información del vector  $\mathbf{v}$  de la siguiente forma: [1er número, cantidad\_de\_ocurrencias\_sucesivas, 2do número, cantidad\_de\_ocurrencias\_sucesivas, ... último número, cantidad\_de\_ocurrencias\_sucesivas]. El vector de entrada puede ser vacío. Se puede utilizar la función de la parte a aunque no la haya realizado.

```

codigo(v) -> resultado

codigo([]) -> []
codigo([0 1 1 1 0]) -> [0 1 1 3 0 1]
codigo([0 0 1 1 1]) -> [0 2 1 3]
codigo([1 1 1]) -> [1 3]
codigo([1 1 1 0 0 1 0 0 1]) -> [1 3 0 2 1 1 0 2 1 1]
codigo([1 0 1 0 1]) -> [1 1 0 1 1 1 0 1 1 1]

```

Parte c)

Dado un vector  $\mathbf{v}$ , que está codificado como en la parte b ([1er número, cantidad\_de\_ocurrencias\_sucesivas, 2do número, cantidad\_de\_ocurrencias\_sucesivas, ... último número, cantidad\_de\_ocurrencias\_sucesivas]), implemente en Octave una función **ITERATIVA decodificar** que devuelva un vector de ceros y unos, en el cual aparezcan la concatenación de cada uno de los números que aparece en  $\mathbf{v}$  repetidos tantas veces como indica su correspondiente cantidad\_de\_ocurrencias\_sucesivas. Esta función corresponde al proceso inverso a la función de la parte b. El vector de entrada puede ser vacío.

```

decodificar(v) -> resultado

decodificar([]) -> []
decodificar([1 3]) -> [1 1 1]
decodificar([0 2 1 3]) -> [0 0 1 1 1]
decodificar([0 1 1 3 0 1]) -> [0 1 1 1 0]
decodificar([1 3 0 2 1 1 0 2 1 1]) -> [1 1 1 0 0 1 0 0 1]
decodificar([1 1 0 1 1 1 0 1 1 1]) -> [1 0 1 0 1]

```

<b>Problema 4</b>	24 (12,12) ptos
-------------------	-----------------

Parte a)

Dada una matriz dispersa **M** en formato elemental **Mf**, **Mc**, **Md**, implemente en Octave la función **RECURSIVA borrarImpares** que devuelve la matriz dispersa **R** en formato elemental que es igual a **M** pero sin los elementos de valor impar.

Parte b)

Dada una matriz dispersa **M** en formato elemental **Mf**, **Mc**, **Md**, implemente en Octave la función **RECURSIVA minMax** que devuelve el mínimo y el máximo de los elementos de la matriz dispersa.

<b>Problema 5</b>	12 Ptos
-------------------	---------

Dada una matriz completa **M** de enteros positivos ( $> 0$ ) y un número **m**, que indica el máximo valor que está almacenado en **M**, implemente en Octave una función **ITERATIVA resumen** que devuelva un vector **v** de largo **m** que en **v(i)** tenga la cantidad de veces que el número **i** está en la matriz **M**. Tenga en cuenta que para poder calcular lo solicitado basta con pasar una única vez por cada posición de la matriz.

**resumen (M,m) -> v**

```
resumen([1 1 3 5 1;3 3 3 3 3;5 1 5 1 1],5)-> [6 0 6 0 3]
resumen([1 1 1 1;1 1 1 1;1 1 1 7],7)-> [11 0 0 0 0 0 1]
resumen([1 1 1 1;1 1 1 1;1 1 1 1],1)-> [12]
resumen([1 2 1 1;2 2 2 1],2)-> [4 4]
```

<b>Problema 6</b>	12 Ptos
-------------------	---------

Dado un vector **v** de números implemente en Octave una función **RECURSIVA simetrica** que devuelva 1 si el vector es simétrico (es decir que si el vector se recorre de izquierda a derecha o de derecha a izquierda se obtienen los mismos elementos) y que devuelva 0 en caso contrario. El vector puede ser vacío.

**simetrica (v) -> resultado**

```
simetrica([])-> 1
simetrica([5])-> 1
simetrica([1 2])-> 0
simetrica([1 1])-> 1
simetrica([1 2 3])-> 0
simetrica([1 2 1])-> 1
simetrica([1 2 2 1])-> 1
simetrica([1 2 2 1 3])-> 0
```