

**Prueba Final - 10 de diciembre de 2020**

- Duración del parcial: 3:00 Hs.
- No se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc). Apague su teléfono celular.
- **Sólo** se contestarán preguntas sobre interpretación de la letra.
- Escriba las hojas de un solo lado. Las partes no legibles del examen se considerarán no escritas.
- En la primera hoja a entregar ponga con letra clara, en el ángulo superior derecho, su **nombre**, número de **cédula de identidad** y **cantidad de hojas** -en ese orden-; en las demás hojas pongan nombre, número de cédula y 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()`
- `floor()`, `ceil()` y `round()`
- `zeros()` y `ones()`

**Nota: - En todos los ejercicios se deben usar las estructuras de control adecuadas para cada caso.  
- Las funciones deben recorrer la cantidad de elementos necesarios para resolver el problema.**

<b>Problema 1</b>	8 (1, 1, 1, 1, 1, 3) ptos	
-------------------	---------------------------	--

- ¿Cuántas representaciones hay del número cero en la representación de signo y magnitud de 8 bits? ¿Cómo se representa?
- ¿Cuál es el menor número que se puede representar con 8 bits en la representación de complemento a 1? ¿Cómo se representa?
- ¿Cuál es el mayor número que se puede representar con 8 bits en la representación de complemento a 1? ¿Cómo se representa?
- ¿Cuál es el menor número que se puede representar con 8 bits en la representación de complemento a 2? ¿Cómo se representa?
- ¿Cuál es el mayor número que se puede representar con 8 bits en la representación de complemento a 2? ¿Cómo se representa?
- En una representación de punto flotante con 1 bit de signo, 4 bits de exponente y 7 bits de mantisa, la representación del número  $\pi$  es: 010001001001. ¿Cuál es la representación de  $\pi/4$ ?

<b>Problema 2</b>	13 (4,9) ptos	
-------------------	---------------	--

- Considere el siguiente código:

```
function a = fRara (b)
    if b == 0
        a = 0;
    elseif mod(b,2) == 0
        a = b + fRara(b/2);
    else
        b = b - 1;
        a = b + fRara(b/2)
    endif
endfunction
```

¿Qué valor queda almacenado en `a` como resultado de la invocación `a = fRara(30)` desde línea de comando?

b) Considere el siguiente código:

```
function b = otraF(a)
    if a == 0
        b = 0;
        c = 1;
    else
        b = a + otraF(a-1);
        c = a * otraF(a-1);
    endif
endfunction
```

¿Qué valor queda almacenado en  $b$ ,  $c$  y  $d$  como resultado de la invocación  $d = \text{otraF}(4)$  desde línea de comando?

<b>Problema 3</b>	10 ptos	
-------------------	---------	--

Implementar en *Octave* la función **iterativa** *sumaDigitos* que, dado un número entero positivo, retorne la suma de sus dígitos.

```
sumaDigitos(16254) → 18
sumaDigitos(1) → 1
sumaDigitos(1899) → 27
```

<b>Problema 4</b>	17 (8,9) ptos	
-------------------	---------------	--

a) Implementar en *Octave* la función **iterativa** *primerCero* que, dado el vector  $x$  que almacena las preimágenes de una cierta función  $f$  de  $\mathbb{R}^+ \rightarrow \mathbb{R}$  y el vector  $fx$  que almacena las imágenes correspondientes (es decir,  $fx(i) = f(x(i))$ ), calcule el valor de la primera preimagen que hace que el valor de la función  $f$  sea 0. Si la función  $f$  no se hace cero, se deberá devolver -1. Asuma que los vectores no pueden ser vacíos.

```
primerCero([1,2,2.5,3,7,9], [1.33,4.36,-1.11,2.34,0,0]) → 7
primerCero([0.47,0.94,1.41,1.88,2.35], [0,1.21,0,1.21,0]) → 0.47
primerCero([1,2,2.5,3.67,4.5], [1.3,2.3,3.65,4.89,6.12]) → -1
```

b) Implementar en *Octave* la función **iterativa** *primerCeroMatriz* que, dado un vector  $x$  que almacena las preimágenes de un conjunto de funciones de  $\mathbb{R}^+ \rightarrow \mathbb{R}$  y la matriz  $Mx$  que almacena en cada fila las imágenes correspondientes a cada una de las funciones, calcule un vector que contenga el valor de la primera preimagen de cada función que hace que el valor funcional sea 0. Para las funciones que no se hacen cero, el valor que se debe almacenar en el vector es -1.

```
primerCeroMatriz([1,2,2.5,3,7], [1.3, 4.3, -1.1, 0, 0;
                                0, 1.21, 0, 1.21, 0;
                                1.3, 2.3, 3.6, 4.8, 6.1]) → [3,1,-1]
```

**Nota:** Para resolver la parte b se puede invocar la función de la parte a.

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

Implementar en *Octave* la función **recursiva** *minMax* que, dado un vector  $x$  de números **no vacío**, retorne el menor y el mayor número del vector.

`minMax([1,2,3,4])` da como resultado 1 y 4.

`minMax([10,2,5,3,-1])` da como resultado -1 y 10.

`minMax([2])` da como resultado 2 y 2.

**Nota:** No se puede utilizar la función `min` ni la función `max`.

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

Implementar en *Octave* la función **recursiva** *min2Vect* que, dados dos vectores  $x$  e  $y$  del mismo largo, retorne un vector  $v$  que en cada posición contenga el menor de los números almacenados en la misma posición de  $x$  e  $y$ . Es decir, para cada posición  $i$  de  $x$  e  $y$ ,  $v(i)=\min(x(i), y(i))$ .

`min2Vect([],[]) = []`

`min2Vect([4],[2]) = [2]`

`min2Vect([4,2,3,1,-2],[3,2,8,-3,-1]) = [3,2,3,-3,-2]`

**Nota:** No se puede utilizar la función `min`.

<b>Problema 7</b>	14 ptos	
-------------------	---------	--

Implementar en *Octave* la función **iterativa** *alMenos* que, dada una matriz  $M$  y dos números *valor* y *cantidad*, retorne 1 si en la matriz existen al menos *cantidad* ocurrencias del número *valor*. En caso contrario, se debe devolver 0.

<b>Problema 8</b>	14 ptos	
-------------------	---------	--

Implementar en *Octave* la función **recursiva** *sumaMatDispRec* que reciba dos matrices dispersas **en formato elemental**, y devuelva el resultado de la suma de las matrices también **en formato elemental**. Considere que los elementos de las matrices están ordenados por fila y luego por columna, es decir, primero van los elementos que corresponden a la fila 1 ordenados por columna, luego los de la fila 2 ordenados por columna, etc. Tenga en cuenta que la suma en una posición puede dar cero y que los ceros no se almacenan en la matriz dispersa.