

- Duración del examen: 3 Hs.
- No se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc). Apague su celular.
- **Sólo** se contestarán preguntas sobre interpretación de la letra hasta 30 minutos antes de la finalización del mismo.
- Las partes no legibles del parcial se considerarán no escritas
- En la primer 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-; las demás hojas es suficiente con 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()`

<b>Problema 1</b>	15 ptos (5,5,5)	
-------------------	-----------------	--

Dados los números 127 y 130:

- Escribalos en punto flotante con la siguiente representación:  
1 bit de signo, 5 bits de exponente y 7 bits de mantisa, siguiendo el criterio de la IEEE para el exponente.
- Realice la suma de ambos números en punto flotante.
- Convierta el resultado de (b) a representación en base 8 y base 10.

<b>Problema 2</b>	28 ptos (5,10,13)	
-------------------	-------------------	--

- Escriba en Octave la función **recursiva** `IdentRc`, que dado un número  $n > 0$ , devuelve la matriz identidad de tamaño  $n \times n$ .
- Escriba en Octave la función **iterativa** `BuscarIt`, que dada una matriz  $A$  de tamaño  $n \times n$ , un número  $X$ , y cuatro enteros `fila1`, `fila2`, `column1` y `column2`, que delimitan una submatriz  $A'$  de  $A$ , devuelva 1 si  $X$  se encuentra dentro de la submatriz  $A'$ , y 0 en caso contrario.  
Asuma que  $1 \leq \text{fila1} \leq \text{fila2} \leq n$  y que  $1 \leq \text{column1} \leq \text{column2} \leq n$ .
- Escriba en Octave la función **iterativa** `ProdIt`, que dada una matriz  $A$   $n \times n$ ,  $n > 0$ , dispersa en formato elemental y un vector columna  $x$  en formato completo y de largo  $n$ , calcula el producto  $v = A * x$ , donde  $v$  también es un vector en formato completo. A modo de repaso de álgebra lineal, el producto  $A * x$  genera un vector columna  $v$ , donde cada elemento  $v(i)$  es igual al producto escalar entre la fila  $i$  de  $A$  y el vector  $x$ .

$$\begin{bmatrix} 1 & 0 & 3 \\ 0 & 2 & 4 \\ 0 & 0 & 9 \end{bmatrix} \begin{bmatrix} 8 \\ 5 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 * 8 + 0 * 5 + 3 * 7 \\ 0 * 8 + 2 * 5 + 4 * 7 \\ 0 * 8 + 0 * 5 + 9 * 7 \end{bmatrix} = \begin{bmatrix} 29 \\ 38 \\ 63 \end{bmatrix}$$

<b>Problema 3</b>	30 ptos (15,15)	
-------------------	-----------------	--

- a) Implemente en Octave una función **iterativa** `ContarIt`, que recibe un vector `T` de largo variable que contiene números enteros con valores entre 1 y 100, y devuelve otro vector `H` con 100 elementos, donde `H(i)` contiene la cantidad de veces que el número `i` aparece en `T`.
- b) Implemente en Octave una función **recursiva** `ContarRc`, que recibe un vector `T` de largo variable que contiene números enteros con valores entre 1 y 100, y devuelve otro vector `H` con 100 elementos, donde `H(i)` contiene la cantidad de veces que el número `i` aparece en `T`.

<b>Problema 4</b>	27 ptos (12,5,10)	
-------------------	-------------------	--

- a) Implemente en Octave una función **recursiva** `Arreglos123Rc`, que dado un número natural  $n > 0$ , devuelve una matriz de dimensión  $n \times 3^n$ , con las  $3^n$  posibles secuencias de largo  $n$  que se pueden formar combinando los dígitos 1, 2 y 3 (una secuencia por columna).
- b) Implemente en Octave una función **iterativa** `Repet123It`, que dado un número natural  $n > 0$ , devuelve un vector fila de largo  $3n$ , compuesto por  $n$  1, seguido por  $n$  2, y culminando con  $n$  3. Por ejemplo el resultado `Repet123It(3)` es el vector: 1 1 1 2 2 2 3 3 3.
- c) Implemente en Octave una función **iterativa** `Arreglos123It`, que dado un número natural  $n > 0$ , devuelve una matriz de dimensión  $n \times 3^n$ , con las  $3^n$  posibles secuencias de largo  $n$  que se pueden formar combinando los dígitos 1, 2 y 3 (una secuencia por columna). Puede utilizar `Repet123It`.

Ejemplos de salida de `Arreglos123It` y `Arreglos123Rc`:

`Arreglos123** (1)`

1 2 3

`Arreglos123** (2)`

1 1 1 2 2 2 3 3 3

1 2 3 1 2 3 1 2 3

`Arreglos123** (3)`

1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3

1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3

1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3