

- 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()`

Problema 1	15 pts (5,5,5)	
-------------------	----------------	--

Dados los números 26 y 31:

- 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 2, base 10 y base 16.

Problema 2	32 pts (11,7,14)	
-------------------	------------------	--

- Escriba en Octave la función **recursiva** `cantFCRec`, que dada una matriz A $n \times n$, $n > 0$, dispersa en formato elemental, calcula el identificador máximo de fila y columna de la matriz (el identificador máximo de fila y columna con coeficiente distinto de 0).
- Escriba en Octave la función **recursiva** `buscarMaxRec`, que dada una matriz A de tamaño $n \times n$, devuelva el coeficiente máximo de la matriz.
- Escriba en Octave la función **iterativa** `maxpFCIt`, que dada una matriz A $n \times n$, $n > 0$, dispersa en formato elemental, y dos enteros `filas` y `columnas` que definen el tamaño de la matriz, calcula dos vectores con los coeficientes máximos por fila y columna de la matriz.

Problema 3	28 pts (16,12)	
-------------------	----------------	--

a) Implemente en Octave una función **iterativa** `comprimirIt`, que recibe un vector `T` de largo variable que contiene números enteros (presumiblemente con muchos números consecutivos repetidos), y devuelve otro vector donde se tiene cada entero que empieza una secuencia y la cantidad de veces que se repite el entero.

```
[4 1 5 1 4 4] = comprimirIt([4 5 4 4 4 4])
```

b) Implemente en Octave una función **recursiva** `descomprimirRec`, que recibe un vector `T` de largo variable que contiene un vector comprimido, y devuelve el vector resultado de descomprimir dicho vector.

```
[4 5 4 4 4 4] = descomprimirRec([4 1 5 1 4 4])
```

Problema 4	25 pts (6, 9, 10)	
-------------------	-------------------	--

a) Implemente en Octave una función **iterativa** `cuantoEstaIt`, que dado un vector `T` de largo variable que contiene números enteros y un número entero `X`, devuelve la cantidad de ocurrencias del número `X` en el vector `T`.

b) Implemente en Octave una función **iterativa** `estaXNIIt`, que dado un vector `T` de largo variable que contiene números enteros, un número entero `X` y un número natural `n`, devuelve 1 si el vector `T` contiene al menos `n` veces el número `X` y 0 en caso contrario.

c) Implemente en Octave una función **recursiva** `estaXNRec`, que dado un vector `T` de largo variable que contiene números enteros, un número entero `X` y un número natural `n`, devuelve 1 si el vector `T` contiene al menos `n` veces el número `X` y 0 en caso contrario.