

2^{do} Parcial - 02 de diciembre de 2015

- 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 hasta 30 minutos antes de la finalización del mismo.
- Escriba las hojas de un solo lado.
- Las partes no legibles del examen se considerarán no escritas.
- En la primer hoja a entregar ponga con letra clara, en el ángulo superior derecho, el salón en el cual desarrolló la prueba, su nombre, número de cédula de identidad y cantidad de hojas -en ese orden-; en las demás hojas es suficiente que ponga nombre, número de cédula y número de página.

Para la resolución de los diferentes ejercicios **solamente** podrá utilizar las siguientes funciones brindadas por **Matlab**:

- `length()` y `size()`
- `mod()` y `rem()`
- `floor()`, `ceil()` y `round()`
- `zeros()` y `ones()`

Problema 1	11 (2,2,3,3,1) pts	
-------------------	--------------------	--

- Expresar en hexadecimal el número 101010101010101_2 .
- Representar el número 4 en complemento a 2 con 5 bits.
- Determinar el número en base 10 que representa la tira $0\ 10000011\ 100000000000000000000000$ en punto flotante de simple precisión de IEEE.
- Escribir el número 3,5 en el sistema de punto flotante de simple precisión de IEEE.
- Ordenar de menor a mayor las siguientes tiras, codificadas en punto flotante de simple precisión de IEEE:
 $1\ 10000011\ 100000000000000000000000$
 $0\ 10000000\ 100000000000000000000000$
 $0\ 10001000\ 000000000000000000000000$

Problema 2	17 (5,6,6) pts	
-------------------	----------------	--

- Escribir una función **iterativa** en Matlab llamada *invertirVector* que devuelva el vector v invertido. Por ejemplo, $\text{invertirVector}([2,4,6,7]) = [7,6,4,2]$, $\text{invertirVector}([\]) = [\]$.
- Escribir una función **recursiva** en Matlab llamada *crearCapicua* que devuelva un nuevo vector w del doble de largo que el vector v , donde la primer mitad sea igual al vector v y la segunda mitad sea igual al vector v invertido. Por ejemplo, $\text{crearCapicua}([2,4,6,7]) = [2,4,6,7,7,6,4,2]$, $\text{crearCapicua}([\]) = [\]$.
- Escribir una función **recursiva** en Matlab llamada *esCapicua* que devuelva 1 si el vector v es capicúa y 0 de lo contrario. Por ejemplo, $\text{esCapicua}([2,4,6,7]) = 0$, $\text{esCapicua}([\]) = 1$, $\text{esCapicua}([2,4,6,7,7,6,4,2]) = 1$, $\text{esCapicua}([2,4,6,7,7,6,4,2]) = 1$.

Problema 3	15 (7,8) ptos		
-------------------	---------------	--	--

- a) Se desea implementar en Matlab una función **iterativa** *Interseccion* que dados dos vectores u y v de números enteros, sin números repetidos y cuyos elementos se encuentran ordenados de menor a mayor, devuelva un vector w que esté formado por los elementos que pertenecen tanto a u como a v (es decir, un elemento pertenece a w , si y sólo si pertenece a u y a v), no tenga elementos repetidos y esté ordenado de menor a mayor.
- b) Se desea implementar en Matlab una función **recursiva** *Diferencia* que dados dos vectores u y v de números enteros, sin números repetidos y cuyos elementos se encuentran ordenados de menor a mayor, devuelva un vector z que esté formado por los elementos que forman parte de u y que no forman parte de v (es decir, un elemento pertenece a z si y sólo si pertenece a u y no pertenece a v), no tenga elementos repetidos y esté ordenado de menor a mayor.

Problema 4	17 (8,9) ptos		
-------------------	---------------	--	--

- a) Implemente en Matlab una función **iterativa** *MaxMinI* que reciba una matriz dispersa en formato elemental y devuelva el máximo valor almacenado en las posiciones pares (celdas de la matriz donde fila+columna da un número par) y el mínimo valor almacenado en las posiciones impares (celdas de la matriz donde fila+columna da un número impar).
- b) Implemente una versión **recursiva** *MaxMinR* de la función de la parte a).

Nota: Recuerde que las matrices dispersas disponen de una gran cantidad de 0s. Esto puede ser tenido en cuenta para calcular el máximo y mínimo.