

Primer Parcial – 26 de setiembre de 2024

- Duración del parcial: 2:30 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 **AMBOS lados**. 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 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()`, `div()` y `rem()`
- `floor()`, `ceil()` y `round()`
- `zeros()` y `ones()`

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 que las necesarias para resolver los problemas.

Problema 1	4 pts	
-------------------	-------	--

Considere el siguiente código en Octave que implementa la función `minimode3` que, dados tres números `n1`, `n2` y `n3` devuelva el menor de los tres números. Complete el código de la función.

```
% minimode3.m
function minimo = minimode3(n1,n2,n3)
    if n1 <= n2 & [ ]
        minimo = n1;
    elseif [ ]
        minimo = n2;
    else
        minimo = n3;
    endif
endfunction
```

Problema 2	4 pts	
-------------------	-------	--

Determine el valor de las variables `a`, `b` y `c` luego de ejecutar `miscript.m` desde la línea de comandos de Octave.

<pre>% func.m function b = func(b,a) b = b*2+a; c = a*2+b; end</pre>	<pre>% miscript.m b = 3; a = 4; a = func(a,b);</pre>
--	--

Problema 3	8 pts	
-------------------	-------	--

Implementar en Octave la función `cambiarUnosCeros` que, dado un vector `v` que contiene números enteros, devuelva un vector del mismo largo sustituyendo los ceros por unos y los unos por ceros, sin cambiar los demás valores.

Ejemplos de ejecución:

```
cambiarUnosCeros([]) → []
cambiarUnosCeros([0,1,3,1,0,2,1]) → [1,0,3,0,1,2,0]
cambiarUnosCeros([1,0,1]) → [0,1,0]
cambiarUnosCeros([2,8]) → [2,8]
```

COMPUTACIÓN 1
Instituto de Computación

Problema 4 8 pts	
---------------------------	--

Implementar en *Octave* la función *primeroMayorQueX* que, dado un vector v de números enteros y positivos, y un número x , devuelva el primer elemento del vector que es mayor que x . En caso de no existir ningún elemento mayor que x en el vector, la función deberá devolver -1.

Ejemplos de ejecución:

```
primeroMayorQueX([], -5) → -1
primeroMayorQueX ([6,10,24,1], -3) → 6
primeroMayorQueX ([6,10,24,1], 10) → 24
primeroMayorQueX ([6,10,24,1], 30) → -1
```

Problema 5 8 pts	
---------------------------	--

Implementar en *Octave* la función *diferencia* que, dados dos vectores $v1$ y $v2$ de números enteros y positivos **ordenados de forma ascendente**, devuelva un vector ordenado con los números enteros que están en $v1$ y no están en $v2$.

Ejemplos de ejecución:

```
diferencia([], [7,9,11]) → []
diferencia([1,2,20,34,41], []) → [1,2,20,34,41]
diferencia([1,2,20,34,41], [7,9,11]) → [1,2,20,34,41]
diferencia([1,2,20,34,41], [1,20]) → [2,34,41]
```

Problema 6 8 pts	
---------------------------	--

Implementar en *Octave* la función *posSecUnosCeros(v)* que, dado un vector v cuyas entradas contienen únicamente 1's y 0's, devuelva dos vectores *posUnos* y *posCeros* con la posición inicial de cada secuencia de 1's y 0's, respectivamente. Si no hay ninguna secuencia de 1's, o 0's, el vector correspondiente es vacío.

Ejemplos de ejecución:

```
posSecUnosCeros([1,1,1,0,0,1,1,1,1,0,1,1,1,1,0,0,0,1,0,1]) →
posUnos=[1,6,11,18,20], posCeros=[4,10,15,19]

posSecUnosCeros([0,0,0,0,0,0,0]) → posUnos=[], posCeros=[1]

posSecUnosCeros([1,1,1,1,1,1,1]) → posUnos=[1], posCeros=[]

posSecUnosCeros([]) → posUnos=[], posCeros=[]
```