

## 1<sup>er</sup> Parcial - 25 de setiembre de 2019

- 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 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 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 alcanza con poner 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>	2 (1, 1) ptos	
-------------------	---------------	--

- Escriba en binario el número  $11010_8$  que está representado en octal. Justifique.
- Escriba en decimal el número  $110_{16}$  que está representado en hexadecimal. Justifique.

<b>Problema 2</b>	2 ptos	
-------------------	--------	--

Escriba en Octave una función `Copa` que reciba una variable **posición**, la cual representa la posición de un equipo en la tabla, y que devuelva la palabra 'Libertadores' si el valor está entre 1 y 4, 'Sudamericana' si está entre 5 y 8 o 'Sin copas' en otro caso.

<b>Problema 3</b>	5 ptos	
-------------------	--------	--

Implementar en *Octave* la función **promedio** que dado un vector  $v$  con al menos un elemento par, devuelva la media de todos sus elementos pares. La media de un conjunto se calcula como la suma de sus elementos dividido por la cantidad de elementos del conjunto.

<b>Problema 4</b>	4 ptos	
-------------------	--------	--

Implementar en *Octave* la función **MayorInd**, que reciba un vector  $v$ , devuelva el mayor índice  $i$  tal que  $v(i+1)$  es 3 veces mayor que  $v(i)$ . En caso de no existir ningún índice que cumpla la condición, devuelva 0. En la medida de lo posible, trate de no leer todo el vector  $v$ .

### Ejemplos:

```
>> MayorInd([2, 3, 5, 5, 2, 15])
ans =
     0
>> MayorInd([2, 3, 5, 15, 3, 15, 45, 10])
ans =
     6
```

<b>Problema 5</b>	6 ptos	
-------------------	--------	--

Implementar en *Octave* la función **SumV**, que reciba un vector  $v$ , devuelva un vector  $s$  tal que  $s(i)$  sea la suma de todos los  $v(k)$  con  $k$  menor o igual a  $i$ .

### Ejemplo:

```
>> y = SumV([2, 3, 1, 5, 2, -5])
y =
     2     5     6    11    13     8
```

<b>Problema 6</b>	12 (4,3,5) ptos	
-------------------	-----------------	--

Considere una matriz de calificaciones  $C$  que en cada fila contiene la cédula sin guión (columna 1) y nota (columna 2) de un estudiante.

- Implementar en *Octave* la función **MinCed**, que reciba una matriz de calificaciones  $C$ , un natural  $ind$  y devuelva el índice de fila del estudiante con menor número de cédula a partir de la fila  $ind$  de  $C$ .
- Implementar en *Octave* la función **Intercambiar**, que dados dos naturales  $i$  y  $j$ , y una matriz  $M$ , devuelva la matriz luego de realizar el intercambio de la fila  $i$  con la fila  $j$ . Asumir que  $i$  y  $j$  son valores válidos de  $M$ .
- Implementar en *Octave* la función **OrdenarHastaCImax**, que reciba una matriz de calificaciones  $C$ , un número de cédula  $ci\_max$ ; y devuelva una matriz de calificaciones con cédula y nota de los estudiantes cuyo valor de cédula es menor que  $ci\_max$ , ordenados por cédula de forma ascendente. La solución debe invocar al menos una de las funciones de las partes a y b aunque no las haya resuelto.

**Ejemplo:**

<p>C =</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>53452361</td><td>1</td></tr> <tr><td>34323425</td><td>9</td></tr> <tr><td>12344234</td><td>12</td></tr> <tr><td>45067207</td><td>4</td></tr> <tr><td>13454345</td><td>5</td></tr> </table>	53452361	1	34323425	9	12344234	12	45067207	4	13454345	5	<pre>&gt;&gt; C2 = OrdenarHastaCImax(C, 45067207) C2 =     12344234    12     13454345     5     34323425     9</pre>
53452361	1										
34323425	9										
12344234	12										
45067207	4										
13454345	5										

<b>Problema 7</b>	9 ptos	
-------------------	--------	--

Dado un vector  $v$  de números, definimos una "secuencia creciente" como un conjunto lo más grande posible de elementos consecutivos tal que para todo  $v(i)$  perteneciente a ese conjunto  $v(i+1) > v(i)$  o  $v(i) > v(i-1)$ . Implementar en *Octave* la función **max\_sec\_crec**, que reciba un vector  $vect$ , y devuelva la mayor de las sumas de todas las "secuencias crecientes" del vector  $vect$ .

**Ejemplo:**

<p>vect = [2,5,7,4,2,7,9,1,2,5,6]</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 50%;">secuencias crecientes</th> <th style="width: 50%;">suma</th> </tr> </thead> <tbody> <tr><td>2,5,7</td><td>14</td></tr> <tr><td>2,7,9</td><td>18</td></tr> <tr><td>1,2,5,6</td><td>14</td></tr> </tbody> </table>	secuencias crecientes	suma	2,5,7	14	2,7,9	18	1,2,5,6	14	<pre>&gt;&gt; vect = [2, 5, 7, 4, 2, 7, 9, 1, 2, 5, 6] &gt;&gt; m = max_sec_crec(vect) m =     18</pre>
secuencias crecientes	suma								
2,5,7	14								
2,7,9	18								
1,2,5,6	14								

**Nota: En todos los ejercicios debe usarse la estructura de control correcta para cada caso.**