

## Prueba Final - 5 de noviembre de 2020

- Duración del parcial: 3:00 Hs.
- No se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc). Apague su teléfono celular.
- 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()`
- `abs()`

Para la resolución de los diferentes ejercicios **solamente** podrá utilizar las siguientes funciones brindadas por **Excel o Libre Office Calc**:

- SUMA
- PROMEDIO
- SI
- Y / O
- BUSCAR / BUSCARV /BUSCARH
- CONTAR

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

Dada la siguiente planilla de cálculo, realizar lo pedido en cada ítem:

	A	B	C	D	E	F
1	Cédula	Nombre	Apellido	Veces en consulta	Pago	
2	15209160	Mayte	Cornejo	3	1482	
3	21969016	Luis	Cañas	1	1411	
4	59575047	Gustavo	Mas	2	1359	
5	32218393	Fernando	Morcillo	2	1453	
6	19382866	Gerard	Figueras	2	1455	
7	63078821	Gertrudis	Molinero	1	1488	
8	58017462	Nadja	Anton	2	1431	
9	20171704	Anton	Orozco	3	1389	
10	64256693	Víctor	Paredes	4	1360	
11	27362285	Rosana	Abad	2	1242	

- a) Escriba la fórmula a transcribir en la celda F2 para hallar la suma de todos los valores en la columna D.
- b) Escriba la fórmula a transcribir en la celda G2 para hallar el promedio de todos los valores en la columna D.
- c) Escriba la fórmula a transcribir en la celda H2 (que luego va a ser copiada al resto de la columna) para devolver la cantidad de consultas de un usuario dividido el total de consultas (cantidad / suma de consultas).

<b>Problema 2</b>	24 (7, 8, 9) pts	
-------------------	------------------	--

Dada la siguiente planilla de cálculo, realizar lo pedido en cada ítem:

	A	B	C
1	Valor 1	Valor 2	Valor 3
2	7	18	20
3	8	11	19
4	5	1	16
5	3	16	3
6	13	0	12
7	6	10	4
8	2	6	5
9	18	10	17
10	2	11	19
11	13	12	12
12	7	2	3
13	10	5	20

- a) Escriba la fórmula a transcribir en la celda D2 (que luego va a ser copiada al resto de la columna) en donde se devuelva “mayor” si el valor2 es mayor que el valor1 de la fila correspondiente. En caso contrario devuelve "menor o igual".
- b) Escriba la fórmula a transcribir en la celda E2 (que luego va a ser copiada al resto de la columna) en donde se devuelva “correcto”, si valor1 es menor que valor2 y valor2 es menor que valor3, de la fila correspondiente. En caso contrario devuelve “incorrecto”.
- c) Escriba la fórmula a transcribir en la celda F2 (que luego va a ser copiada al resto de la columna) en donde se devuelva el valor más grande entre valor1, valor2 y valor3 de la fila correspondiente.

<b>Problema 3</b>	12 (6, 6) pts	
-------------------	---------------	--

Los siguientes fragmentos de código no funcionan como deberían o no usan la estructura correcta. Escriba el código correcto de cada parte.

- a) Se quiere realizar la suma de los elementos de un vector (una lista):

```
n=length(v);
suma=0;
for i<=n
    suma= suma + v(i);
end
```

- b) Se quiere guardar z=1, cuando b es menor a y:

```
while b<y
    z=1;
end
```

<b>Problema 4</b>	12 pts	
-------------------	--------	--

Implementar en *Octave* la función **clasificar** que dado un número  $X$ , devuelva 1 si el valor es mayor a 3 o menor a -3, y devuelva 0 en caso contrario.

Ejemplos:

```
>> y=clasificar(5)
y= 1
```

```
>> y= clasificar(1)
y= 0
```

```
>> y= clasificar(-10)
y= 1
```

```
>> y= clasificar(-1.5)
y= 0
```

<b>Problema 5</b>	17 pts	
-------------------	--------	--

Implementar en *Octave* la función **minV** que dado un vector  $v$  con al menos un elemento, devuelva el mínimo valor del vector.

Ejemplo:

```
>> minimo= minV([2 3 6 -2 6])
minimo= -2
```

<b>Problema 6</b>	20 pts	
-------------------	--------	--

Implementar en *Octave* la función **buscar** que dado un vector  $v$  y un valor  $val$ , devuelva 1 si el vector contiene ese valor y 0 en caso contrario.

Ejemplos:

```
>> res=buscar([3 2 6 8],9)
res= 0
```

```
>> res=buscar([2 6 8 10],8)
res=1
```

<b>Nota: En todos los ejercicios debe usarse la estructura de control correcta para cada caso.</b>
--