

## Primer Parcial – 22 de setiembre de 2022

- Duración del parcial: 2:00 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.
- Puede escribir 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 (2, 2) pts

En este ejercicio `X`, `Y` y `Z` son variables escalares (no vectores) booleanas (sus valores son 1 o 0).

a) Reescribir la siguiente expresión lógica sin utilizar el operador `&` (o `&&`):

$$r = (x \ \&\& \ y) \ || \ (\sim x \ \&\& \ \sim y)$$

b) Reescribir la siguiente expresión lógica sin utilizar el operador `&` (o `&&`):

$$r = (x \ \&\& \ y \ \&\& \ (z \ || \ \sim z)) \ || \ (\sim x \ \&\& \ \sim y \ \&\& \ (z \ || \ \sim z))$$

**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.

```
% func1.m
function [c,b] = func1(b,a)
    c = 1;
    n = b;
    for a = 1:n
        c = c + b;
    end
end
```

```
% miscript.m
c = 3;
a = 4;
[b,c] = func1(a,a);
a = a + a;
```

**Problema 3** | 6 pts

Implementar en *Octave* la función `cantidadCalificaciones` que, dado un vector `x` que contiene las notas correspondientes a calificaciones en la escala del 1 al 12 de cada uno de los estudiantes de un curso, devuelve un vector con la cantidad de veces que aparece cada una de las notas (es decir, en la posición 1 estará la cantidad de estudiantes que obtuvieron 1, en la 2 los que obtuvieron 2, etc).

`cantidadCalificaciones ([2,1,2,1,10]) → [2,2,0,0,0,0,0,0,0,1,0,0]`

<b>Problema 4</b>	11(3,2,6) pts	
-------------------	---------------	--

Representaremos una medida de tiempo como un vector de tres elementos [h,m,s] donde h, m y s son enteros que representan horas, minutos y segundos, respectivamente; siendo tanto m como s valores entre 0 y 59.

a) Implementar en *Octave* la función *sumarTiempos* que, dado dos vectores *t1* y *t2* representando dos medidas de tiempo, devuelva un vector *suma\_t* representando el resultado de sumar *t1* y *t2*.

```
sumarTiempos([0,0,0], [0,0,0]) → [0,0,0]
sumarTiempos([4,3,35], [10,30,4]) → [14,33,39]
sumarTiempos([4,50,35], [10,9,36]) → [15,0,11]
sumarTiempos([14,50,3], [13,19,36]) → [28,9,39]
```

b) Implementar en *Octave* la función *horaEnPunto* que, dado un vector *t* representando una medida de tiempo, devuelva 1 si la medida de tiempo representa una cantidad de horas exactas o en punto (es decir si los minutos y segundos son 0). En caso contrario devuelve 0.

```
horaEnPunto([0,0,0]) → 1
horaEnPunto([14,0,0]) → 1
horaEnPunto([4,11,0]) → 0
horaEnPunto([1,0,22]) → 0
```

c) Implementar en *Octave* la función iterativa *buscarHoraEnPunto* que, dada una matriz H, donde cada fila es una medida de tiempo, devuelva el índice de la primera hora exacta o en punto (índice más bajo) en la matriz H. De no haber ninguna hora en punto la función devuelve -1. Nota: Se puede utilizar la función de la parte b aunque no la haya implementado.

<pre>H = [1, 3, 1;       5, 1, 7;       6, 4, 6]; buscarHoraEnPunto(H) → -1</pre>	<pre>H = [1, 3, 1;       5, 0, 0;       6, 0, 0]; buscarHoraEnPunto(H) → 2</pre>
---	--

<b>Problema 5</b>	7 pts	
-------------------	-------	--

Implementar en *Octave* la función *aciertos* que, dados dos vectores **ordenados de forma ascendente** *vApuesta*, que contiene los cinco números de una jugada del 5 de Oro, y *vSorteo*, que contiene los cinco números que resultaron sorteados en el 5 de Oro, devuelva la cantidad de aciertos que obtuvo la jugada en el sorteo.

```
aciertos([1,2,20,34,41], [3,6,7,8,10]) → 0
aciertos([1,2,20,34,41], [1,6,7,20,25]) → 2
aciertos([1,2,5,13,17], [1,5,7,20,25]) → 2
aciertos([1,2,5,13,17], [1,2,5,13,17]) → 5
```

<b>Problema 6</b>	8 pts	
-------------------	-------	--

Implementar en *Octave* la función *reemplazarEnMatriz* que, dada una matriz *Mx* que contiene números y dos valores *in* y *out*, devuelva una matriz con los mismos elementos que *Mx*, pero en la cual cada ocurrencia del valor *in* se reemplaza por el valor *out*.

<pre>Mx = [2, 4, 1, 3, 1;       3, 1, 5, 1, 7;       3, 3, 6, 4, 6];</pre>	<pre>reemplazarEnMatriz(Mx,3,8) → [2, 4, 1, 8, 1;  8, 1, 5, 1, 7;  8, 8, 6, 4, 6];</pre>
--	--