

Segundo Parcial – 28 de noviembre de 2023

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.

Escriba las hojas de los dos 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 ejercicios **solamente** podrá utilizar las siguientes funciones de **Octave**:

- `length()`, `size()` y `isempty()`
- `mod()` y `rem()`
- `fix()`, `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 o invocaciones recursivas que las necesarias para resolver el problema

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

En todas las siguientes partes deben justificarse los resultados:

a) Exprese en exceso M de 4 bits ($M=7$) el resultado de $1001 + 0011$, que también están en exceso M.

b) Halle un número x en Comp a 1 de 8 bits, tal que $10000010 + x$ da overflow en dicha representación, e indique por qué.

c) Halle un número x en Comp a 2 de 8 bits, tal que $11110001 - x$ da overflow en dicha representación, e indique por qué.

d) Exprese en hexadecimal el resultado de $EFF34A * (F + 1)$.

e) En aritmética de punto flotante con 1 bit de signo, 4 bits de exponente y 7 bits de mantisa, donde el

exponente se representa con desplazamiento $M=7$: dados dos números normalizados $a=(s_a, e_a, m_a)$ y $b=(s_b, e_b, m_b)$, exprese la condición que debe darse para que $a+b=a$.

Problema 2	5 pts	
-------------------	-------	--

Considere la siguiente función:

```
function lista2 = func(lista)
    n = length(lista);
    if n==0
        lista2 = [0];
    elseif mod(lista(1),2)==0
        lista2 = [lista(1), func(lista(2:n))];
        lista = [lista(2:n), lista(1)];
    else
        lista2 = [func(lista(2:n)), lista(1)];
        lista = [lista(1), lista(2:n)];
    end
```

¿Qué valor queda almacenado en z, n y lista como resultado de las siguientes invocaciones?:

```
lista = [1,2,3,4];
z = func(lista);
```

Problema 3	10 pts	
-------------------	--------	--

Considere tres matrices T , P y V del mismo tamaño, que en la posición (i,j) contienen el dato de temperatura, presión y volumen, respectivamente, de una misma muestra gaseosa. El número de moles N se calcula como $(P*V)/(R*T)$, siendo R la constante de los gases ideales 8,314 J/mol.K. Implementar en *Octave* la función **iterativa** *maximoN*, que, dadas las tres matrices, T , P y V , devuelva la muestra (i,j) que maximiza N . Nota: Puede asumir que existe al menos una muestra, es decir que las matrices son no vacías.

Problema 4	10 pts	
-------------------	--------	--

Implementar en *Octave* la función **recursiva** *sumaParesImpares* que dado un vector v retorna dos valores *sumaI* y *sumaP*, donde *sumaI* es la suma de los valores de v que se encuentran en posiciones impares y *sumaP* es la suma de los valores de v que se encuentran en posiciones pares.

```

sumaParesImpares([]) → sumaI = 0, sumaP = 0
sumaParesImpares([2]) → sumaI = 2, sumaP = 0
sumaParesImpares([2,1,-3]) → sumaI = 2-3 = -1, sumaP = 1
sumaParesImpares([2,3,6,4,3,1]) → sumaI = 2+6+3 = 11, sumaP = 3+4+1 = 8

```

Problema 5	14 (6,8) pts	
-------------------	--------------	--

En la red social Y cada usuario tiene un identificador numérico entero. Las relaciones posibles entre dos usuarios son: seguir (representada con un 1), bloquear (representada con un -1) o nula. Estas relaciones se almacenan en tres vectores: *usrA*, *usrB*, *relac*, de forma que, si el usuario x sigue al usuario y , existirá algún entero i tal que $usrA(i)=x$, $usrB(i)=y$, $relac(i)=1$. De la misma forma, si x bloquea a y , la relación se representa mediante $usrA(i)=x$, $usrB(i)=y$, $relac(i)=-1$. Finalmente, si x tiene una relación nula con y (es decir, si x no sigue ni bloqueó a y), no se almacena ningún dato en los vectores. Nota: los tres vectores tienen el mismo largo.

Ejemplo:

$usrA = [1 \ 1 \ 1 \ 3 \ 2]$	El usuario 1 sigue al usuario 2 y bloqueó a los usuarios 3 y 4. El usuario 2
$usrB = [2 \ 3 \ 4 \ 1 \ 3]$	sigue al usuario 3 y tiene relación nula con los usuarios 1 y 4. El usuario 3
$relac = [1 \ -1 \ -1 \ 1 \ 1]$	sigue al usuario 1 y tiene relación nula con los usuarios 2 y 4.

a) Escriba en *Octave* una función **recursiva** *siguiendo*, que dado un identificador de usuario u , y las relaciones de la red Y representadas por los tres vectores descritos, devuelva un vector con los identificadores de los usuarios **a los que sigue** el usuario u .

b) Escriba en *Octave* una función **recursiva** *desbloquear*, que dado un identificador de usuario $u1$, otro $u2$, y las relaciones de la red Y representadas por los tres vectores descritos, devuelva el conjunto de las relaciones de la red resultantes luego de que $u1$ desbloquea a $u2$ (pasando al estado de relación nula).

Problema 6	10 pts	
-------------------	--------	--

Implementar en *Octave* la función **recursiva** *restarValor* que dadas la matriz M con números enteros mayores o iguales a 0, la matriz *casillas* de dos columnas que representa una lista de casillas y el número *valor* que es un entero mayor o igual a cero, devuelva la matriz que resulta de restar el *valor* de las posiciones indicadas en la lista de casillas de la matriz M . La matriz resultante también debe contener números enteros mayores o iguales a 0, es decir que, en aquellas casillas que resultan negativas se pone un 0 en su lugar.

Ejemplos:

$M = [4 \ 3 \ 2 \ 1;$	$restarValor([2,2;2,1;3,3;1,4],M,2) \rightarrow$	$[4 \ 3 \ 2 \ 0;$
$3 \ 2 \ 1 \ 0;$		$1 \ 0 \ 1 \ 0;$
$2 \ 0 \ 1 \ 0;$		$2 \ 0 \ 0 \ 0;$
$1 \ 0 \ 0 \ 1]$		$1 \ 0 \ 0 \ 1]$

$restarValor([2,2;2,1;3,3;1,4],M,0) \rightarrow M$