

**Segundo Parcial – Modalidad Presencial - 30 de noviembre de 2021**

- ☪ 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 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 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() 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**

<b>Problema 1</b>	10 (1, 1, 1, 1, 3, 3) pts	
-------------------	---------------------------	--

En todos los siguientes ejercicios deben justificarse los resultados:

- a) Convierta el número  $11111110_2$  a un número entero en base 10.
- b) Calcule el resultado de  $1000010 + 1000001$  en complemento a 1 de 8 bits. ¿Es el resultado representable en dicho sistema?
- c) Calcule el resultado de  $11110001 + 00001111$  en complemento a 2 de 8 bits. ¿Es el resultado representable en dicho sistema?
- d) Represente el número  $11101000111_2$  en sistema octal.
- e) Complete la siguiente función que devuelve un vector con los dígitos del número n en base 10 representado en la base que se pasa por parámetro:

```
function v = ent10_a_base(n, base)
    v=[];
    while 0 < n
        v=...;
        n=...;
    end
endfunction
```

- f) En una representación de punto flotante con 1 bit de signo, 4 bits de exponente y 7 bits de mantisa, la representación del número pi es: 010001001001. ¿Cuál es la representación de  $\pi*8$ ?

**Problema 2** 6 pts

Considere el código de las siguientes dos funciones:

```
function a = f1 (b)
    if b == 0
        a = 3;
    elseif mod(b,2) == 0
        a = b+f2(floor(b/2));
    else
        b = b - 1;
        a = b+f1(floor(b/2));
    endif
endfunction
```

```
function b = f2(a)
    if a == 1
        b = 1;
        d = 0;
    else
        d = f1(a-2);
        b = 2*a + d;
        d = b-a + d;
    endif
endfunction
```

¿Qué valor queda almacenado en  $a$ ,  $b$ ,  $d$  y  $z$  como resultado de las siguientes invocaciones  
 $b = 9$ ;  
 $z = f1(b)$ ;  
 desde línea de comando?

**Problema 3** 8 pts

Implementar en *Octave* la función **recursiva recurrente** que, dado un número  $n$  entero mayor que 0, calcule la siguiente recurrencia:

$$\begin{cases} a_1=1 \\ a_n=a_{n-1}+n*n \end{cases}$$

recurrente(1) → 1  
 recurrente(2) → 5  
 recurrente(3) → 14  
 recurrente(4) → 30

<b>Problema 4</b>	13 pts
-------------------	--------

Implementar en Octave la función **recursiva** *perteneceRecorrida* que dado:

- una matriz *recorrida* de dos columnas que representa un conjunto de puntos del plano por los que pasa una partícula.
- una coordenada  $x$
- una coordenada  $y$ ;

devuelva el índice ( $i$ ) de la fila de la matriz *recorrida* correspondiente a la posición  $(x,y)$ , en caso de que la partícula pase por dicha posición, es decir, si  $(x,y)$  está en *recorrida*. Si la partícula no pasa por  $(x,y)$ , la función devuelve -1. Se puede asumir que en *recorrida* no hay posiciones repetidas.

Ejemplos de ejecución (los ejemplos siguen el formato *perteneceRecorrida*(*recorrida*,  $x$ ,  $y$ )  
→  $i$ )

```
recorrida = [2.1 2.9;
            3.1 4.5;
            4.2 6.3;
            5.2 4.3;
            7.2 5.1;
            6.4 7.3]
```

```
perteneceRecorrida(recorrida, 4.2, 6.3) → 3 (ya que la fila 3 es [4.2 6.3])
perteneceRecorrida(recorrida, 6.4, 6.3) → -1 (ya que [6.4 6.3] no está en
recorrida)
```

<b>Problema 5</b>	10 pts
-------------------	--------

Implementar en Octave la función **iterativa** *minMaxMatriz* que, dada una matriz de números no vacía, devuelva el menor y el mayor número de la matriz.

*minMaxMatriz*([2]) da como resultado 2 y 2, respectivamente.

*minMaxMatriz* ([1,2,3,4]) da como resultado 1 y 4, respectivamente.

*minMaxMatriz* ([1,2,3,4; -1, 10, 9, 0]) da como resultado -1 y 10, respectivamente.

*minMaxMatriz* ([2,3,4; 6,7,9; 0,8,8; 1,1,1; 3,9,5]) da como resultado 0 y 9, respectivamente.

Nota: No se puede utilizar la función *min* ni la función *max*.

<b>Problema 6</b>	13 pts
-------------------	--------

Implementar en Octave la función **recursiva** *paresMatDispRec* que reciba una matriz dispersa **en formato elemental**, y devuelva otra matriz **en formato elemental** conteniendo solamente los elementos con valor numérico par.