

**2<sup>do</sup> Parcial - Diciembre de 2004**

- Duración de esta etapa: 3 Hs
- No se podrá utilizar ningún tipo de material (apuntes, libro, calculadora, etc).
- **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 primer hoja a entregar ponga con letra clara, en el ángulo superior derecho, salón en el cual desarrolló la prueba, su nombre, número de cédula de identidad y cantidad de hojas -en ese orden-; las demás hojas es suficiente con nombre, número de cédula y número de página.
- Al entregar su prueba recuerde firmar la planilla correspondiente

<b>Problema 1</b>	15 pts	
-------------------	--------	--

- a) Expresar el número  $-13$  en 5 bits, en las siguientes representaciones (si no se puede representar indíquelo)
- signo y valor absoluto
  - Exceso a M (el enfoque que utiliza la IEEE)
  - complemento a 1
  - complemento a 2
- b) Siendo los números  $A = 11110$  y  $B = 01100$  que están codificados en complemento a 2 de 5 bits, indique el resultado de las siguientes operaciones  $A + B$  y  $A - B$  (si no se puede representar indíquelo).
- c) Indique que números representan las siguientes codificaciones en punto flotante (IEEE):
- 00111111001000000000000000000000
  - 00000000010000000000000000000000
- d) Codifique en punto flotante de simple precisión (IEEE) el número 31

<b>Problema 2</b>	15 pts	
-------------------	--------	--

- a) Diferencia Simétrica entre conjuntos

Sean  $A$  y  $B$  conjuntos, La *diferencia simétrica* entre  $A$  y  $B = \{x / (x \in A \vee x \in B) \wedge x \notin A \cap B\}$

Ejemplo: Si  $A = [2\ 4\ 6\ 9]$  y  $B = [2\ 5\ 6\ 8]$  entonces  $DS = [4\ 5\ 8\ 9]$

Escriba una **función recursiva** *DifSimetrica* en **Matlab** que reciba como entrada dos conjuntos ordenados de números enteros sin repetir y devuelva el conjunto ordenado *diferencia simétrica* de los dos anteriores. La función debería sacar provecho de que los conjuntos de entrada están ordenados.

- b) Escriba una **función iterativa** *Union* en **Matlab**, que reciba como entrada dos conjuntos ordenados de números enteros sin repetir y devuelva el conjunto ordenado sin repetidos que resulta de la unión de los anteriores. **La función no debe recorrer más de una vez cada vector, esto incluye los vectores de entrada, salida y cualquier otro auxiliar que utilice.**

Ejemplo: Si  $A = [2\ 4\ 6\ 9]$  y  $B = [2\ 5\ 6\ 8]$  entonces  $Union(A, B) = [2\ 4\ 5\ 8\ 9]$

---

**Nota:** En este ejercicio NO se pueden usar las funciones `sort` ni `find` de Matlab o cualquier otra que, por su naturaleza, resuelva trivialmente el problema.  
Todos los vectores que se utilicen, deben ser recorridos como máximo una vez.

---

<b>Problema 3</b>	15 pts	
-------------------	--------	--

Existe un juego llamado MasterMind. En dicho juego, un jugador elige un número de  $n$  cifras y otro jugador debe adivinar ese número probando sucesivamente cada vez con un nuevo número hasta adivinar. Cada vez que el jugador que está adivinando propone un nuevo número, el otro le informa cuan cerca estuvo de adivinar. Para ello, le dice:

- Cuantos dígitos adivinó y están en la posición correcta.
- Cuantos dígitos adivinó pero no están en la posición correcta.

Si el número a adivinar es "1234" y el segundo jugador prueba con el número "4230" el resultado es 2 dígitos correctos y en la posición correcta (el 2 y el 3) y 1 un dígito correcto pero en otra posición (el 4).

Escriba una **función iterativa** *MasterMind* en **Matlab** que recibe como parámetro el **número** a adivinar y el **número** propuesto, ambos con la misma cantidad de dígitos y **sin dígitos repetidos**, y devuelve un vector  $y$ , donde:

$y(1)$  = Dígitos correctos y en la posición correcta.

$y(2)$  = Dígitos correctos pero en otra posición.

Ejemplo :  $\text{MasterMind}(1234, 4230) = [2 \ 1]$

$\text{MasterMind}(1254, 4235) = [1 \ 2]$

---

**Nota:** En este ejercicio NO se puede usar la función `find`, `sort` ( o similares ) de Matlab o cualquier otra que, por su naturaleza, resuelva trivialmente el problema.

---

<b>Problema 4</b>	15 pts	
-------------------	--------	--

Dados dos números enteros  $N$  y  $B$ , donde  $B$  es menor que  $N$ , se puede hacer que  $N$  "explote" usando a  $B$  como bomba.

Cuando  $N$  explota se parte en dos números enteros  $N1 = N \text{ div } B$  y  $N2 = N - (N \text{ div } B) \cdot B$ , donde  $(N \text{ div } B)$  es el cociente de la división entera de  $N$  entre  $B$

Una explosión produce una reacción en cadena, esto quiere decir que si  $N1$  o  $N2$  son mayores que  $B$ , también explotarán.

Por ejemplo, sean  $N = 10$  y  $B = 3$

- $N1 = 10 / 3 = 3$ , No vuelve a explotar, por ser igual a  $B$
- $N2 = 10 - (10/3) \cdot 3 = 1$ , Vuelve a explotar por ser mayor que  $B$

La explosión se detiene cuando todos los enteros obtenidos como resultado de las sucesivas explosiones sean menores o iguales a  $B$ . De esta forma se logra partir el número  $N$  en pedazos menores o iguales a  $B$ .

Para el caso del ejemplo el número 10 se parte en 3, 2, 1, 1 y 3

Escriba una **función recursiva** *explotar* en **Matlab** que recibe como parámetro los números  $N$  y  $B$  y devuelve un vector con los enteros resultantes de la explosión de  $N$  usando a  $B$  como bomba.

Ej:  $\text{explotar}(10, 3) = [3 \ 2 \ 1 \ 1 \ 3]$

$\text{explotar}(5, 2) = [2 \ 1 \ 2]$

$\text{explotar}(3, 2) = [1 \ 2]$

---

**Nota:** En este ejercicio NO se puede usar ninguna función de Matlab que por su naturaleza, resuelva trivialmente el problema.

---