



2^{do} Parcial - Diciembre de 2006 - 2^a parte

- Duración de esta etapa: 2 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 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

Problema 1	20 ptos (10,10)	
-------------------	-----------------	--

El método de bipartición para buscar un entero dado, dentro de un vector de enteros, **ordenado** en forma **ascendente**, se basa en la siguiente idea:

Sea el entero n el valor a buscar dentro del vector v

Se compara el entero n con el entero almacenado en el medio del vector v

si $n = v(\text{medio})$, entonces el entero pertenece al vector;

si $n > v(\text{medio})$, entonces se debe buscar el entero en la segunda mitad del vector v ;

si $n < v(\text{medio})$, entonces se debe buscar el entero en la primera mitad del vector v ;

Continuando este proceso se puede o bien encontrar el elemento o bien concluir que no pertenece al vector

Se pide:

- Implemente una **función iterativa biparticionI**(n,v) en **Matlab** que resuelva el método de bipartición. Debe devolver 1 si el elemento pertenece al vector y 0 en caso contrario.
- Implemente una **función recursiva biparticionR**(n,v) en **Matlab** que resuelva el método de bipartición. Debe devolver 1 si el elemento pertenece al vector y 0 en caso contrario.

Nota: Se permite utilizar la función **floor** o **ceil** de **Matlab**.

En este ejercicio NO se permite utilizar ninguna función de Matlab que, por su naturaleza, resuelva trivialmente el problema.

Problema 2	10 ptos	
-------------------	---------	--

Se llaman **números amigos** a los números naturales que cumplen que la suma de los divisores de uno de ellos nos da el otro.

Un ejemplo de números amigos son: 220 y 284

Los divisores de 220 son: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110

Los de 284 son: 1, 2, 4, 71 y 142.

Si sumamos todos los divisores de 220: $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$

Se pide realizar una **función iterativa numerosAmigos** en **Matlab** que reciba como parámetro dos números enteros y devuelva 1 si son amigos o 0 en caso contrario.

Nota: Se permite utilizar las funciones **rem** o **mod** y **floor** de **Matlab**.

En este ejercicio **NO** se puede utilizar la función **sum**, ni ninguna función **Matlab** que, por su naturaleza, resuelva trivialmente el problema.

Se permite utilizar funciones auxiliares pero deben ser implementadas por usted

Problema 3 | 20 pts (10,10)

- a) Escribir una **función recursiva** *reverso* en **Matlab**, que toma como parámetro un vector v y devuelve un vector con los mismos elementos pero en orden inverso.

Ejemplo: $\text{reverso}([2 \ 5 \ 6 \ 1 \ 3 \ 7]) = [7 \ 3 \ 1 \ 6 \ 5 \ 2]$

- b) Escribir una **función recursiva** *rotar* en **Matlab**, que toma como parámetro una matriz M y devuelve una matriz con sus mismos elementos rotados 180° .

Ejemplo: $\text{rotar} \begin{pmatrix} 4 & 3 & 2 & 2 \\ 2 & -1 & 2 & -3 \\ 7 & 6 & 5 & 3 \end{pmatrix} = \begin{pmatrix} 3 & 5 & 6 & 7 \\ -3 & 2 & -1 & 2 \\ 2 & 2 & 3 & 4 \end{pmatrix}$

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