

Examen - Julio de 2009 - 2ª parte

- Duración del examen: 3 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 20 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.

Problema 1	23 (11,12) ptos	
-------------------	-----------------	--

a) Escriba una función **suma_por_filas** en *Matlab* que tome como entrada una matriz (representación completa) de tamaño $m \times n$ y devuelva un vector de m elementos, donde el elemento en la posición i del vector sea la suma de las celdas de la fila i de la matriz.

Ejemplos:

`suma_por_filas` ($\begin{bmatrix} 4 & 1 & 3 \\ 2 & -2 & 1 \\ -3 & 1 & -2 \end{bmatrix}$) devolvería $[8 \ 1 \ -4]$

`suma_por_filas` ($\begin{bmatrix} -1 & 0 & -2 \\ 3 & 0 & 0 \end{bmatrix}$) devolvería $[-3 \ 3]$

b) Escriba una función **suma_filas_disp** (A_s, A_i, A_j, m) en **MatLab** que reciba como parámetros una matriz dispersa almacenada en formato elemental (representada mediante **tres vectores**) y la cantidad de filas de la matriz (m), y devuelva un vector de m elementos, donde el elemento en la posición i del vector sea la suma de las celdas de la fila i de la matriz.

Notas: Para la resolución de este ejercicio no se permite el uso de la función `sum` u otras funciones *Matlab* que permitan resolver de manera trivial el problema. Puede asumir que en la representación dispersa utilizada para la parte b no contendrá ninguna celda cuyo elemento valga 0.

Problema 2	19 ptos (6,7,6)	
-------------------	-----------------	--

- Implementar una función recursiva que verifica si 2 vectores son iguales (mismos elementos en el mismo orden).
- Implementar una función iterativa que verifica si 2 vectores son iguales (mismos elementos en el mismo orden).
- Implementar una función recursiva que recibe 2 vectores y verifica si uno es prefijo del otro.

Nota: En este ejercicio NO se permite el uso de la función de *Matlab* que por su naturaleza, resuelvan trivialmente el problema.

Problema 3	28 (14,14) ptos	
-------------------	-----------------	--

El método de bipartición para encontrar las raíces de una ecuación $f(x) = 0$, donde f es una función continua se basa en la siguiente idea:

Si a y b son dos puntos tales que $f(a) < 0 < f(b)$, entonces f tiene un cero entre a y b .

Para buscarlo, sea $c = (a+b)/2$.

si $f(c) = 0$, entonces f tiene un cero en c ;

si $f(c) > 0$, entonces f tiene un cero entre a y c ;

si $f(c) < 0$, entonces f tiene un cero entre c y b .

Repitiendo este proceso se encuentra la raíz de $f(x) = 0$, con la precisión deseada (i.e. hasta que $|a-b|$ sea menor que un número dado).

Se pide:

- Implemente una **función recursiva biparticionR**(a,b,err,f) en **Matlab** que resuelva el método de bipartición.
- Implemente una **función iterativa biparticionI**(a,b,err,f) en **Matlab** que resuelva el método de bipartición.

Observaciones:

- Utilice el comando de matlab **feval** para evaluar la función en el valor c . Por ejemplo para evaluar la función `sin` de Matlab en el valor 1, se tiene que invocar **feval('sin',1)**. Para el cabezal de ejemplo se tiene que invocar a **feval(f,c)**.
- Asuma que la función que se le pasa como parámetros posee un cero en el intervalo y cumple que $f(a) < 0 < f(b)$.
- En caso que el intervalo sea menor que el error, la función debe retornar el extremo menor.

Nota: Se permite utilizar las funciones **abs** y **feval**.

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