



2^{do} Parcial - 05 de Diciembre de 2007 - 2^a parte

- 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.

Problema 1	18 ptos (7,11)	
-------------------	----------------	--

Parte a)

Realizar una **función recursiva** *SumaPol(v,p)* en *Matlab* que recibe dos polinomios en formato *Matlab* y calcula la suma de los mismos.

Parte b)

Realizar una **función iterativa** *MultiPol(v,p)* en *Matlab* que recibe dos polinomios en formato *Matlab* y calcula la multiplicación de los mismos.

Obs.

Las funciones tienen que poder operar con polinomios de distintos grados. En el caso de la suma, si se anulan coeficientes no es necesario eliminar los coeficientes 0 del comienzo.

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

Problema 2	16 ptos (8,8)	
-------------------	---------------	--

Considere la función **pos_max: Matriz** $\rightarrow (x,y)$. Dicha función toma como entrada una matriz, busca el elemento que tenga valor máximo y devuelve la posición de dicho elemento. Por ejemplo, para la matriz:

$$A = \begin{bmatrix} -1 & 8 & 9 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & -2 \end{bmatrix}$$

El resultado sería (1, 3), ya que el lugar (1, 3) de la matriz tiene el valor 9. Notar que se debe devolver primero la fila y luego la columna.

Parte a)

Dada la representación elemental de matrices dispersas vista en clase, escriba una **función iterativa** *pos_max* en *Matlab*, que tome como parámetro una matriz dispersa y devuelva como resultado la posición en la matriz que corresponda al valor máximo de la misma.

Parte b)

Escriba una **función recursiva** en *Matlab* que tome como parámetro una matriz dispersa y devuelva como resultado la posición de la matriz que corresponda al valor máximo de la misma.

Obs:

- * Puede asumir que existirá un sólo elemento con el valor máximo de la matriz.
- * La representación elemental de matrices dispersas utilizada es con tres vectores.

Nota: En este ejercicio **NO** se pueden utilizar las funciones **sum**, **max**, ni ninguna función **Matlab** que, por su naturaleza, resuelva trivialmente el problema.

Problema 3	16 ptos (8,8)
-------------------	---------------

El sistema-L de Lindenmayer para modelar el crecimiento de algas se basa en la siguiente sucesión.

Comienza con un vector que contiene la representación de un alga en su primera etapa de vida, o sea un vector con un uno ($[1]$). En cada intervalo de tiempo t , para el crecimiento de la misma, cada valor 1 cambia por el par de números $1, 2$ dentro del vector; y cada valor 2 cambia por el valor 1 .

Las primeras etapas del crecimiento son las siguientes:

$[1]$ crece en un intervalo de tiempo t a: $[1, 2]$

$[1, 2]$ crece en un intervalo de tiempo t a: $[1, 2, 1]$

$[1, 2, 1]$ crece en un intervalo de tiempo t a: $[1, 2, 1, 1, 2]$

$[1, 2, 1, 1, 2]$ crece en un intervalo de tiempo t a: $[1, 2, 1, 1, 2, 1, 2, 1]$,

$[1, 2, 1, 1, 2, 1, 2, 1]$, y así continúa el crecimiento.

Parte a)

Implementar una **función recursiva reproduceAlga** en **Matlab**, que dado un alga (un vector) nos devuelva el alga (el vector) resultante del crecimiento de un único intervalo de tiempo t .

Parte b)

Implementar una **función recursiva crecimientoDeAlga** en **Matlab**, que dado un alga (un vector) y un entero n mayor a cero, devuelva el alga (el vector) después de n crecimientos de la misma (n intervalos de tiempo t).

Nota: En este ejercicio **NO** se pueden utilizar funciones **Matlab** que, por su naturaleza, resuelvan trivialmente el problema.
