# CC1001-7 Auxiliar Extra – Control 3Arreglos, Computación Numérica y MATLAB

Daniel Calderon – Sebastián Fehlandt
3 de Junio de 2009

### P1) Búsqueda y Ordenamiento(30min)

a) Implemente una función de encabezado

public static int[] buscar(Comparable x, Comparable[][] y)

que retorne la posición {i,j} del elemento x en la matriz y. Suponga que la matriz esta ordenada de menor a mayor hacia la derecha, y luego hacia abajo.

b) Implemente una función de encabezado

public static int ordenarSegun(Comparable[] x,Comparable[] y)

que ordene “de menor a mayor” los arreglos x e y según las comparaciones realizadas con x. Y retorne el número de líneas de los arreglos. Utilícela en un programa que muestre una lista de pares (apellido\_alumno, notas), ordenándola alfabéticamente según el apellido del alumno.

c) (Propuesto) Implemente una función de encabezado

public static Object[][] revolver(int[] x, int[] y, Object[][] m)

que retorne un arreglo de Object donde el elemento i,j corresponda al elemento m[x[i]][y[j]].

### P3) Complejos y Matriz de Complejos (Propuesto)

Implemente la clase complejo con los siguientes métodos:

|  |  |
| --- | --- |
| **private** **double** real,imag; | Partes real e imaginaria del complejo |
| **public** Complejo(**double** x,**double** y) | Constructor |
| **public** **double** real() | Retorna parte real del complejo |
| **public** **double** imag() | Retorna la parte imaginaria  |
| **public** **double** abs() | Retorna el módulo del complejo |
| **public** **double** ang() | Retorna el ángulo del complejo |
| **public** Complejo conjugado() | Parte imaginaria cambia de signo |
| **public** Complejo sumar(Complejo c) |  |
| **public** Complejo restar(Complejo c) |  |
| **public** Complejo multiplicar(**double** r) | Retorna el producto por escalar del complejo |
| **public** Complejo multiplicar(Complejo c) |  |
| **public** Complejo invertir() | $$z^{-1}=\frac{\overline{z}}{\left|z\right|^{2}}$$ |
| **public** Complejo dividir(Complejo c) |  |

Implemente la Clase MatrizCompleja con los mismos métodos de la clase matriz pero definida por las variables de instancia:

protected Complejo[][] x; protected int n,m;

### P2) Clase Matriz (30min)

Implementa la clase Matriz con los siguientes métodos:

|  |  |
| --- | --- |
| **protected** **double**[][] x;**protected** **int** n,m; | Variables de instancia |
| **public** Matriz(**int** n,**int** m) | Construye una matriz de nxm sin asignar valores |
| **public** Matriz(**int** n,**int** m,**double** v) | Construye una matriz de nxm asignando el valor v a todas las celdas |
| **public** Matriz(**double**[][] x) | Construye una matriz en base al arreglo x |
| **public** **int**[] dim() | Dimensión de la matriz. Ej: {n,m} |
| **public** **boolean** equalsDim(Matriz w) | True si tienen igual dimensión. |
| **public** **void** set(**double** d,**int** i,**int** j) | Asigna el valor “d” a la componente i,j de la Matriz |
| **public** **double** get(**int** i,**int** j) | Retorna la componente i,j de la Matriz |
| **public** Matriz traspuesta() | Retorna la matriz traspuesta |
| **public** Matriz getSubmatriz(**int** i0,**int** j0,**int** i1,**int** j1) | Entrega una submatriz formada desde la celda i0,j0 hasta la celda i1,j1 |
| **public** Matriz setSubmatriz(**int** i0,**int** j0,Matriz w) | Modifica los valores de la submatriz a partir de i0,j0 poniendo los valores de la matriz w |
| **public** Matriz ponderar(**double** a) | Retorna una matriz ponderada en un factor a  |
| **public** Matriz sumar(Matriz w) | Retorna la suma de ambas matrices  |
| **public** Matriz multiplicar(Matriz w) | Retorna la matriz producto de ambas |
| **public** **void** print() | Imprime la matriz, entre elementos de la fila use “\t” y entre filas use “\n” |
| **public** **static** Matriz rand(**int** n,**int** m) | Genera una matriz de nxm con números aleatorios entre 0 y 1 |
| **public** **static** Matriz rand(**int** n) | Genera una matriz de nxn con números aleatorios entre 0 y 1 |
| **public** **static** Matriz id(**int** n) | Genera la matriz identidad de dimensión n |

### P4) MATLAB (45min)

a) Escriba la función de encabezado function out=u(t)que retorna el resultado de aplicar la función escalón unitario a t (que puede ser un numero o un vector). La función escalón unitario se define como:

$$u\left(t\right)=\left\{\begin{array}{c}1 si t\geq 0\\0 si no\end{array}\right.$$

b) Escriba la función de encabezado function out=rect(t) que retorna el resultado de aplicar la función rect a t (que puede ser un numero o un vector). La función rect se define como:

$$rect\left(t\right)=\left\{\begin{array}{c}1 si t\in [-0.5 , 0.5]\\0 si no \end{array}\right.$$

c) Escriba la función function out = derivar(t,x) donde t y x corresponden a vectores de la misma longitud cuyos elementos se relacionan de la siguiente manera x(i) = f(t(i)), donde f es alguna función, es decir, los pares ordenados (t(i),x(i)) corresponden a puntos descritos por la función f en el plano cartesiano. La función debe retornar para cada índice i, la función derivada de f evaluada en dicho punto, mediante una aproximación por secantes.

d) Escriba la función:

function out = integrarTrapecios(t,x) donde t y x corresponden a los mismos vectores que en la parte anterior. La función debe retornar para cada índice i, la función primitiva de f evaluada en dicho punto, es decir, para cada índice i, debe retornar el área bajo la curva entre el índice 0 y el índice i.

e) Escriba una función que reciba los mismo parámetros que las 2 anteriores y grafique (con subplot) la función, su derivada y su primitiva.

f) Escriba un programa que pida al usuario el nombre de una función (m-file), intervalo a considerar y número de puntos y realice los mismo gráficos que en la parte anterior.