

## Trabajo Práctico N° 4

### Subrutinas – Apuntadores

En cada uno de los ejercicios siguientes, a medida que avance, reutilice las funciones creadas en ejercicios anteriores cuando sea posible

1. Escriba una función que reciba como argumento un entero  $N$ , cree un vector de  $N$  elementos de tipo double dinámicamente (utilizando la función malloc), y devuelva un apuntador con la dirección de memoria del arreglo creado
2. Escriba una función que reciba como parámetros 2 vectores  $\mathbf{u}$  y  $\mathbf{v}$  de  $N$  elementos tipo double (donde  $N$  puede ser variable), cree un nuevo vector  $\mathbf{w}$  de  $N$  elementos dinámicamente, y calcule (almacenando el resultado en  $\mathbf{w}$ ) el producto vectorial  $\mathbf{u} \times \mathbf{v}$ . La función debe devolver el apuntador a  $\mathbf{w}$  (que fue creado en la misma función).
3. Escriba una función que reciba un vector de  $N$  elementos double (donde  $N$  puede ser variable) y un escalar double  $s$ , y escale el mismo arreglo utilizando el factor  $s$  (es decir, no debe devolver un puntero ni crear otro arreglo)
4. Escriba una función que reciba como argumento un vector de  $N$  elementos double (donde  $N$  puede ser variable) y que genere valores aleatorios para el mismo. Asegurese de generar valores con cifras decimales (es decir que no sean simplemente enteros).
5. Escriba una función que reciba un vector de  $N$  elementos ( $N$  variable) e imprima por pantalla el vector en un formato legible para el ser humano (ej: utilizando notación algebraica: " $(v_1, v_2, \dots, v_n)$ ")
6. Escriba un programa que:
  - a. lea por teclado un entero  $N$ ,
  - b. cree 4 vectores de  $N$  elementos double dinámicamente con valores aleatorios,
  - c. lea por teclado un factor de escala double
  - d. escale 2 de los vectores generados en (b) por el factor  $s$
  - e. calcule el producto vectorial de los vectores de a pares, el primero por el segundo y el tercero por el cuarto, generando como resultado 2 vectores  $\mathbf{u}$  y  $\mathbf{v}$
  - f. calcule el producto vectorial de  $\mathbf{u} \times \mathbf{v}$
  - g. muestre todos los resultados intermedios por pantalla
  - h. libere la memoria dinámica utilizada
7. Repita el ejercicio (1) para el caso de una matriz de  $M \times N$  elementos
8. Escriba una función que reciba como parámetros 2 matrices  $\mathbf{A}$  y  $\mathbf{B}$  de  $M \times N$  y  $N \times P$  elementos tipo double, respectivamente, cree una nueva matriz  $\mathbf{C}$  de  $M \times P$  elementos dinámicamente, y calcule (almacenando el resultado en  $\mathbf{C}$ ) el producto matricial. La función debe devolver el apuntador a  $\mathbf{C}$  (que fue creada en la misma función). Valide los tamaños de las matrices
9. Repita el ejercicio (3) para el caso de una matriz de  $M \times N$  elementos
10. Repita el ejercicio (4) para el caso de una matriz de  $M \times N$  elementos
11. Repita el ejercicio (5) para el caso de una matriz de  $M \times N$  elementos
12. Escriba una función que reciba como parámetro una matriz de  $M \times N$  elementos, y calcule la transpuesta de la matriz. El resultado debe ser almacenado en la misma matriz (es decir que no se debe reservar memoria para el resultado ni devolver ningún apuntador)

13. Escriba un programa que:

- a. Lea por teclado 3 valores M, N y P
- b. Cree 4 matrices dinámicamente con valores aleatorios:
  - i. 2 de M x N y
  - ii. 2 de N x P
- c. Lea por teclado un factor de escala double
- d. Escale las 4 matrices generadas en (b) por el factor s
- e. Calcule el producto matricial de a pares, generando 2 matrices C<sub>1</sub> y C<sub>2</sub> de MxP elementos cada una
- f. Calcule la transpuesta de C<sub>2</sub>
- g. Calcule el producto de C<sub>1</sub> x C<sub>2</sub><sup>T</sup> (MxP y PxM respectivamente)
- h. Muestre los resultados intermedios por pantalla
- i. Libere la memoria dinámica utilizada

14. La Sucesión de Fibonacci viene dada por las siguientes ecuaciones:

$$F_n = F_{n-1} + F_{n-2}$$

$$F_0 = 1$$

$$F_1 = 1$$

Escriba una función recursiva que reciba como argumento un entero **n**, y que calcule la sucesión hasta el valor F<sub>n</sub>