

Trabajo Práctico N° 3 - A

Parte 1 – Estructuras de datos y cadenas

1. Escriba un programa que:
 - a. Genere un vector de tamaño N con valores aleatorios enteros, utilizando la función rand(). Use la función srand() para obtener una mejor aleatoriedad.
 - b. Recorra el vector y guarde el promedio de los valores en una variable
 - c. Encuentre el menor de los elementos
2. Implementar una función que genere una matriz con todos los elementos en 0 excepto aquellos para los que $i+j$ sea par; para estos elementos generar un valor aleatorio entre 1 y 10000.
3. Escriba un programa que lea una cadena con una operación matemática básica de dos términos (suma, resta, multiplicación y división) y calcule el resultado. Por ej. el usuario ingresa "2.4+3", el resultado a devolver es "5.4". Considere los datos de entrada números flotantes. Considere una cadena máxima de 100 caracteres.
4. Escriba un programa que lea una frase y evalúe en forma recursiva si la misma corresponde a un palíndromo. Considere una cadena máxima de 100 caracteres.
5. Modifique el ejercicio 12 del práctico anterior (el de las fechas) utilizando un struct que represente una fecha completa. Realice al menos una función que reciba por parámetro una fecha. Defina la estructura fecha como un tipo definido por el usuario (typedef)
6. Escriba un programa que calcule el producto vectorial de 2 vectores en R^3 utilizando structs para representar los vectores

Parte 2 - Apuntadores, memoria dinámica y estructuras dinámicas

7. Implemente un programa que: (a) declare una variable v de tipo int; (b) llame a una función void duplicar(int *v) pasándole como parámetro la dirección de memoria de la variable v (mediante &v); (c) la función debe multiplicar v por 2 y colocar el resultado en la misma dirección de memoria, de manera que el programa main "vea" el cambio de valor; (d) imprima por pantalla el valor de la variable v para verificar que el programa funciona correctamente.
8. 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.
9. 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)
10. Escriba un programa que, reutilizando las dos funciones anteriores.
 - a. lea por teclado un entero N,
 - b. cree 2 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
 - a. muestre todos los resultados
 - b. Libere la memoria dinámica utilizada
11. Escriba una función que reciba como parámetros 2 matrices **A** y **B** de $M \times N$ y $N \times P$ elementos tipo double, respectivamente, cree una nueva matriz **C** de $M \times P$ elementos dinámicamente, y calcule (almacenando el resultado en **C**) el producto matricial. La función debe devolver el apuntador a **C** (que fue creada en la misma función). Valide los tamaños de las matrices para que se pueda realizar la operación.
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, utilizando las dos funciones anteriores:
- a. Lea por teclado 3 valores M , N y P
 - b. Cree 2 matrices dinámicamente con valores aleatorios:
 - i. La primer matriz (C_1) de $M \times N$
 - ii. La segunda matriz (C_2) de $P \times N$
 - a. Calcule la transpuesta de la segunda matriz (C_2), dando lugar a la matriz C_{2T}
 - c. Calcule el producto de $C_1 \times C_{2T}$
 - d. Muestre los resultados intermedios por pantalla
 - e. Libere la memoria dinámica utilizada