

Perea Rojas Giovanni, conclusiones ejercicios 1-6

Ejercicio 1:

El código estuvo genial para practicar C: manejamos memoria dinámica, ordenamos datos con el `struct Alumno` y guardamos la info. Usar `realloc` fue un acierto para que el programa creciera sin problemas. En resumen, el ejercicio unió memoria dinámica, estructuras y manejo de archivos.

Ejercicio 2: Este código se trató de aprender a leer y usar los datos que teníamos guardados en `alumnos.txt`, pasándolos a un arreglo de structs. lo que nos costó un poco fue darle a esta info en la memoria, recorriendo el arreglo para todo lo solicitado.

Ejercicio 3:

Lo que más nos costó fue la persistencia de datos: guardar (`fprintf`) y cargar (`fscanf`) el inventario desde/hacia `inventario.txt`.

El menú y la búsqueda (`strcmp`) integraron el proyecto, demostrando la colaboración entre memoria dinámica y manejo de archivos para crear un programa que recuerda la información.

Ejercicio 4:

La dificultad en este ejercicio radicó en implementar la asignación de memoria en dos pasos (punteros para filas, luego memoria para columnas). La clave fue la disciplina: validar cada `malloc` y liberar en orden inverso para evitar fugas y asegurar una destrucción segura.

Ejercicio 5

El proyecto de clasificación de datos se centró en mejorar la lectura de archivos (cambiando `fscanf` por `fgets` y `strtok` para separar datos por comas: grupo, nombre, calificación). Con los datos limpios, se usó lógica anidada para clasificar alumnos por grupo, acumular calificaciones y contar miembros, finalizando con el cálculo del promedio grupal. Fue una práctica completa de lectura, procesamiento y análisis estadístico de archivos.

Ejercicio 6:

Este proyecto simuló una base de datos real.

Las funciones de "Alta" y "Consulta" reforzaron el manejo de arreglos dinámicos y la búsqueda.

La lección crucial fue la "Eliminación": aprendimos que borrar un registro en `clientes.txt` requiere reescribir el archivo completo omitiendo el dato, asegurando la persistencia y gestionando el ciclo de vida completo del dato (creación, lectura y destrucción segura).