



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Programación Estructurada

Actividad 10:

FUNCIONES y METODOS DE ORDENACION Y BUSQUEDA

ESTRUCTURAS Y LIBRERIAS

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 15 de Octubre del 2023

Introducción:

En el mundo de la programación, las funciones y métodos de ordenación y búsqueda desempeñan un papel fundamental. Estas herramientas nos permiten organizar y buscar datos de manera eficiente en una variedad de aplicaciones, desde procesamiento de información hasta desarrollo de software. Además, el uso de estructuras y librerías adecuadas es esencial para optimizar nuestros programas y hacer que el código sea más comprensible.

En esta actividad, exploraremos en detalle el uso de funciones y métodos de ordenación y búsqueda, así como el empleo de estructuras y librerías para lograr un código más eficiente y legible. También destacaremos la importancia de utilizar typedef y struct para hacer que nuestras variables y estructuras sean menos tediosas de manejar.

Competencia:

En esta actividad, nos enfocamos en mejorar nuestras habilidades en el uso de funciones y métodos de ordenación y búsqueda. Aprenderemos a aplicar estructuras y librerías de manera efectiva, lo que nos ayudará a optimizar nuestros programas y mantener la eficiencia original de C. Además, exploraremos la utilidad de typedef y struct para hacer que nuestro código sea más comprensible y fácil de mantener.

La competencia radica en nuestra capacidad para aplicar estos conceptos de manera inteligente y lógica, creando soluciones que sean eficientes y evitando la sensación de que simplemente copiamos y pegamos código.

Fundamentos:

1. Funciones de ordenación y búsqueda: Las funciones de ordenación nos permiten organizar datos en una secuencia específica, como orden alfabético o numérico. Las funciones de búsqueda nos ayudan a encontrar elementos particulares en conjuntos de datos, acelerando el proceso de recuperación de información relevante.

2. Estructuras y librerías: Las estructuras nos permiten agrupar diferentes tipos de datos bajo un solo nombre, lo que facilita la organización y el acceso a la información. Las librerías proporcionan conjuntos de funciones predefinidas que podemos utilizar en nuestros programas para realizar tareas comunes de manera eficiente.

3. Typedef y struct: La palabra clave `typedef` nos permite definir alias más legibles para tipos de datos, lo que hace que nuestras declaraciones de variables sean más comprensibles. Por otro lado, `struct` nos permite crear estructuras personalizadas para organizar datos relacionados en un solo bloque.

4. Validación: La validación es esencial para garantizar que nuestros programas funcionen de manera confiable y no se rompan debido a datos incorrectos o mal formateados. Los métodos de validación son críticos para mantener la integridad de los datos y la robustez del software.

Funciones de ordenación y búsqueda:

Ordenación: La ordenación es el proceso de organizar un conjunto de datos en un orden específico, como numérico o alfabético, para facilitar la recuperación eficiente de información. Algunos algoritmos de ordenación comunes incluyen el algoritmo de burbuja, el algoritmo de inserción, el algoritmo de selección y algoritmos más avanzados como el Merge Sort y el Quick Sort. La elección del algoritmo de ordenación adecuado depende del tamaño de los datos y de los recursos disponibles.

Búsqueda: La búsqueda se refiere a la acción de encontrar un elemento específico en un conjunto de datos. Existen varios algoritmos de búsqueda, como la búsqueda binaria, la búsqueda lineal y la búsqueda hash. La elección del algoritmo de búsqueda adecuado depende de la organización de los datos y del tiempo que se requiera para buscar un elemento.

Estructuras y librerías:

Estructuras de datos: Las estructuras de datos son elementos fundamentales en programación. Permiten organizar datos de manera lógica y eficiente. Algunos ejemplos comunes de estructuras de datos incluyen arrays, listas enlazadas, pilas, colas, árboles y grafos. La elección de la estructura de datos adecuada depende de la tarea a realizar y de los requisitos de rendimiento.

Bibliotecas estándar: Las bibliotecas estándar, como la biblioteca estándar de C o las bibliotecas de otros lenguajes de programación, proporcionan una amplia gama de funciones y estructuras de

datos predefinidas para realizar tareas comunes. Estas bibliotecas ahorran tiempo y esfuerzo al programador, ya que pueden aprovechar las implementaciones probadas y optimizadas que ofrecen.

Typedef y struct:

Typedef: La palabra clave ``typedef`` se utiliza para crear alias para tipos de datos existentes. Esto hace que el código sea más legible y facilita la adaptación a cambios futuros en los tipos de datos. Por ejemplo, ``typedef int Entero`` permite usar ``Entero`` en lugar de ``int`` en declaraciones de variables, lo que hace que el código sea más comprensible.

Struct: ``struct`` es una construcción en C que permite definir estructuras personalizadas que contienen varios campos de datos relacionados. Las estructuras son útiles para agrupar datos relacionados bajo un solo nombre y permiten crear tipos de datos complejos. Por ejemplo, se puede definir una estructura ``Persona`` con campos como ``nombre``, ``edad`` y ``dirección`` para representar a una persona en un programa.

Validación:

Validación de datos: La validación de datos es el proceso de asegurarse de que los datos proporcionados cumplan con ciertos criterios o restricciones antes de ser procesados. Esto ayuda a evitar errores y problemas en el programa. Por ejemplo, en una aplicación de registro de usuarios, se podría validar que la dirección de correo electrónico sea válida y que la contraseña cumpla con ciertos requisitos de seguridad.

Validación de entrada: La validación de entrada se utiliza para garantizar que los datos ingresados por el usuario sean correctos y seguros. Esto es fundamental para prevenir ataques de seguridad, como la inyección de SQL. La validación de entrada puede incluir la verificación de datos numéricos, fechas, direcciones de correo electrónico, contraseñas y otros datos sensibles.

Procedimiento:

ACTIVIDAD 10

REALICE EL SIGUIENTE PROGRAMA QUE CONTENGA UN MENÚ.

MENÚ

- 1.- AGREGAR (AUTOM 10 REGISTROS)
- 2.- AGREGAR MANUAL
- 3.- ELIMINAR REGISTRO (lógico)
- 4.- BUSCAR
- 5.- ORDENAR
- 6.- IMPRIMIR
- 0.- SALIR

UTILIZAR UN ARREGLO DE 500 REGISTROS

SE DEBERÁ **UTILIZAR ESTRUCTURAS** CON LOS DATOS BÁSICOS DE UN ALUMNO (status, Matricula, ApPat, ApMat, Nombre, Edad, Sexo)

Busqueda y Ordenacion por campo MATRICULA

nota: usar librería propia

Conclusiones:

En esta actividad, hemos explorado en profundidad las funciones y métodos de ordenación y búsqueda, así como el uso de estructuras y librerías en la programación. Hemos aprendido cómo utilizar ``typedef`` y ``struct`` para hacer que nuestro código sea más legible y eficiente. Además, hemos destacado la importancia de la validación en la programación para evitar problemas y errores. En última instancia, el objetivo de esta actividad es que como programadores, podamos aplicar estos conceptos de manera inteligente y lógica, optimizar nuestro código y desarrollarnos constantemente. Al hacerlo, no solo creamos programas eficientes, sino que también mejoramos nuestras habilidades y conocimientos en programación, lo que nos permite enfrentar desafíos más complejos con éxito.

Anexo:

<https://github.com/BrayanArM98/Estructurada/blob/main/ACTIVIDAD%2010%20ESTRUCTURADA.pdf>