



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en computación

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Kevin Alejandro Gonzalez Torres

Matrícula: 372354

Maestro: Pedro Núñez Yépiz

Actividad Numero: 13

Tema - Unidad: Estructuras

Ensenada Baja California a 15 de octubre del 2022



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. INTRODUCCIÓN

En el lenguaje de programación C, las estructuras son una característica fundamental que permite a los programadores definir tipos de datos personalizados para organizar y almacenar múltiples variables relacionadas bajo un solo nombre. Estas estructuras son una forma eficiente de representar objetos complejos en un programa, ya que permiten agrupar datos de diferentes tipos en una sola entidad. En esencia, una estructura en C es una colección de variables de diferentes tipos de datos (como enteros, flotantes, caracteres, punteros, etc.) que se agrupan bajo un nombre común. Cada variable en una estructura se denomina "miembro" o "campo". Estos campos pueden tener nombres significativos que facilitan la manipulación de los datos.

La palabra clave typedef crea un alias "Punto" para la estructura "struct Punto", lo que facilita la declaración y uso de variables de tipo "Punto" en el código. Esta práctica implica la creación de un programa en el lenguaje de programación C que tenga la capacidad de generar el Identificador Único de Registro de Población (CURP) de cualquier individuo al ingresar únicamente los datos esenciales, tales como nombre, apellidos, fecha de nacimiento, entidad federativa y sexo. Este programa debe estar equipado para manejar excepciones y verificar la validez de los datos ingresados, asegurando así la integridad y precisión de la CURP generada. En resumen, las estructuras en C son una forma esencial de organizar y manipular datos complejos, y typedef es una característica útil para simplificar la declaración y uso de estructuras en el código.

2. COMPETENCIA

- Programación en C
- Diseño de estructuras de datos
- Manipulación de datos complejos
- Uso de typedef
- Validación y manejo de excepciones
- Desarrollo de software que cumple con requerimientos específicos
- Integridad y precisión de datos



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

3. FUNDAMENTOS

Cadenas:

<https://www.programiz.com/c-programming/c-strings>

Structs:

<https://www.geeksforgeeks.org/structures-c/>

Funciones:

https://www.w3schools.com/c/c_functions.php

Typedef:

<https://www.geeksforgeeks.org/typedef-in-c/>

<https://learn.microsoft.com/es-es/cpp/c-language/typedef-declarations?view=msvc-170>

Struct

https://www.w3schools.com/c/c_structs.php

<https://www.geeksforgeeks.org/structures-c/>



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

4. PROCEDIMIENTO

ACTIVIDAD 13

REALICE EL SIGUIENTE PROGRAMA QUE CONTENGA UN MENÚ.

MENÚ

- 1.- AGREGAR (AUTOM 100 REGISTROS)
- 2.- EDITAR REGISTRO
- 3.- ELIMINAR REGISTRO (lógico)
- 4.- BUSCAR
- 5.- ORDENAR
- 6.- IMPRIMIR
- 7.- GENERAR ARCHIVO TEXTO
- 8.- VER ARCHIVO TEXTO
- 9.- CREAR ARCH BINARIO
- 10.- CARGAR ARCH BINARIO
- 11.- MOSTRAR ELIMINADOS
- 0.- SALIR

UTILIZAR UN ARREGLO DE 5000 REGISTROS

SE DEBERÁ UTILIZAR ESTRUCTURAS CON LOS DATOS BÁSICOS DE UN EMPLEADO

preguntar nombre de archivo binario o de archivo texto

Busqueda y Ordenacion por CAMPO LLAVE

Nota 1: usar librería propia con funciones

Nota 2: 100 % validado, Cuidar desbordamiento de vector

Nota 3: Campo llave matricula no repetido, archivos solo cargar 1 sola vez.

Nota 4: Usar el tipo Tkey para hacer más practico el programa



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

5. RESULTADOS Y CONCLUSIONES

El código muestra una implementación funcional de un sistema de gestión de registros de trabajadores. Tiene aspectos positivos, como la eficiencia en las búsquedas, la variedad de operaciones ofrecidas y el uso de persistencia de datos en archivos binarios. Además, la generación aleatoria de datos para los registros de prueba es una característica práctica.

- **main** (función principal): Inicia el programa y llama a la función menu para gestionar las operaciones del usuario.
- **msges** (menú de opciones): Muestra un menú al usuario, le permite seleccionar una opción y devuelve la opción validada.
- **menu** (manejo del menú): Implementa un bucle que gestiona las operaciones del usuario según la opción seleccionada en el menú.
- **autoDataReg** (generación de datos aleatorios): Crea y devuelve una estructura Tworker con datos aleatorios para nuevos registros de trabajadores.
- **linearSearch** (búsqueda lineal): Busca un número de registro en el arreglo y devuelve su índice si se encuentra, -1 si no.
- **binarySearch** (búsqueda binaria): Realiza una búsqueda binaria de un número de registro en el arreglo ordenado y devuelve su índice si se encuentra, -1 si no.
- **selectionSort** (ordenamiento por selección): Ordena el arreglo por el número de registro utilizando el algoritmo de selección.
- **swap** y **partition** (quicksort): Funciones auxiliares para el algoritmo de ordenamiento quicksort.
- **quicksort** (ordenamiento rápido): Ordena el arreglo por el número de registro utilizando el algoritmo quicksort.
- **printRegister** (impresión de registros): Imprime en la consola los registros del arreglo en formato tabular.
- **writeTextFile** (escritura en archivo de texto): Escribe los registros en un archivo de texto con formato tabular.
- **printTextFile** (lectura y impresión de archivo de texto): Lee y muestra en consola el contenido de un archivo de texto.
- **createBinaryFile** (escritura en archivo binario): Escribe los registros en un archivo binario.
- **readBinaryFile** (lectura de archivo binario): Lee registros desde un archivo binario y los carga en el arreglo.
- **printDeleted** (impresión de registros eliminados): Imprime en formato tabular los registros marcados como eliminados.

6. ANEXOS

GTKA_RP13_PE_ANEXOS



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

7. REFERENCIAS

Diseño de algoritmos y su codificación en lenguaje C

Corona, M.A. y Ancona, M.A. (2011)..

España: McGraw-Hill.

ISBN: 9786071505712

Programación estructurada a fondo: implementación de algoritmos en C

:Pearson Educación. Sznajdleder, P. A. (2017)..

Buenos Aires, Argentina: Alfaomega

Como programar en C/C++

H.M. Deitel/ P.J. Deitel

Segunda edición

Editorial: Prentice Hall.

ISBN: 9688804711

Programación en C. Metodología, estructura de datos y objetos

Joyanes, L. y Zahonero, I. (2001)..

España: McGraw-Hill.

ISBN: 8448130138