



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: Eliel Alfonso Ontiveros Ojeda

Matrícula: 368746

Maestro: Pedro Núñez Yépiz

Actividad No. : 8

Tema - Unidad : Teoría Arreglos y Funciones

Ensenada Baja California a 01 de octubre del 2023



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. INTRODUCCIÓN

La teoría de arreglos y funciones es un tema fundamental en la programación y las ciencias de la computación. Los arreglos son estructuras de datos que nos permiten almacenar y manipular grandes cantidades de información de manera eficiente, mientras que las funciones nos permiten encapsular bloques de código que realizan tareas específicas, lo que facilita la reutilización de código y la organización de programas complejos.

2. COMPETENCIA

Al estudiar la teoría de arreglos y funciones, se adquieren las siguientes competencias:

1. Comprender y aplicar los conceptos básicos de arreglos y funciones.
2. Ser capaz de declarar, inicializar y manipular arreglos en varios lenguajes de programación.
3. Entender cómo se almacenan los arreglos en la memoria.
4. Ser capaz de definir y llamar a funciones, entender la diferencia entre argumentos y parámetros, y cómo se pasa la información entre funciones.
5. Aplicar estos conceptos en la resolución de problemas de programación.

3. FUNDAMENTOS

Los arreglos son estructuras de datos que contienen elementos del mismo tipo. Cada elemento en un arreglo tiene un índice, que es un número que especifica su posición en el arreglo.

Las funciones, por otro lado, son bloques de código que realizan una tarea específica. Una función puede tomar argumentos (datos de entrada), realizar operaciones en estos argumentos y luego devolver un resultado (datos de salida).



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

4. PROCEDIMIENTO

Realiza programa en C el programa deberá tener el siguiente menú.

MENÚ

- 1.- LLENAR **VECTOR 1** (MANUALMENTE)
- 2.- LLENAR **VECTOR 2** ALEATORIAMENTE
- 3.- LLENAR **VECTOR 3** (CON VECTOR1 Y VECTOR2)
- 4.- IMPRIMIR VECTORES
- 5.- LLENA MATRIZ 4 X 4
- 6.- IMPRIMIR MATRIZ
- 0.- SALIR

NOTA: EL PROGRAMA DEBERÁ REPETIRSE CUANTAS VECES LO DESEE EL USUARIO

NOTA 2: EL **VECTOR 1** DE 10 POSICIONES, NÚMEROS DEL **30 AL 70**

NOTA 3: EL **VECTOR 2** DE 10 POSICIONES CON NÚMEROS GENERADOS ALEATORIAMENTE DEL 1 AL 20 (SIN REPETIR)

NOTA 4: EL **VECTOR 3** DE 20 POSICIONES, CON LOS DATOS DEL ARREGLO1 Y ARREGLO2

NOTA 5: MATRIZ 4 X 4 LLENARLA CON LOS DATOS DEL VECTOR1 Y VECTOR2

5. RESULTADOS

1. **Declaraciones de funciones:** Estas son las funciones que realizan las tareas principales del programa. Incluyen:
 - void manual(int vect1[10]): Esta función pide al usuario que introduzca manualmente 10 números entre 30 y 70 en el vector 1.
 - void random(int vect2[10]): Esta función genera 10 números aleatorios entre 1 y 20 y los almacena en el vector 2.
 - void vect1_vect2(int vect1[10], int vect2[10], int vect3[20]): Esta función combina el vector 1 y el vector 2 en un tercer vector (vector 3).
 - void imp_vectores(int vect1[10], int vect2[10], int vect3[20]): Esta función imprime los tres vectores.
 - void matriz_4x4(int matrix[30][30], int vect1[10], int vect2[10]): Esta función selecciona los primeros 4 números de cada vector para crear una matriz 4x4.
 - void imp_matriz(int matrix[30][30]): Esta función imprime la matriz 4x4.
2. **Función Principal:** La función int main() es el punto de entrada del programa. Llama a la función menu() que controla el flujo del programa en función de la entrada del usuario.
3. **Función de Menú:** La función void menu() muestra un menú al usuario y llama a la función apropiada en función de la elección del usuario. Utiliza una declaración de switch-case para determinar qué función llamar.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

6. CONCLUSIONES

En conclusión, la teoría de arreglos y funciones es esencial para cualquier persona que quiera convertirse en un programador competente. Estos conceptos forman la base de muchos algoritmos y estructuras de datos más avanzados, y entenderlos a fondo puede abrir la puerta a una amplia gama de oportunidades en el campo de la informática. Aunque puede ser desafiante al principio, con práctica y estudio, cualquier persona puede dominar estos conceptos fundamentales.

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



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño