



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Solano Meza Angel Daniel

Matrícula: 372453

Maestro: Pedro Núñez Yépiz

Actividad No. : 8

Tema - Unidad : Arreglos - Unidad 1

Ensenada Baja California a 30 de Septiembre del 2023



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. INTRODUCCIÓN

En esta práctica se trabaja con arreglos bidimensionales y unidimensionales, estos son una colección de datos del mismo tipo que comparten el nombre y se identifican mediante un índice. Para poder trabajar con arreglos debemos dominar los ciclos y condiciones, ya que los arreglos integran estos mismos para el poder ingresar valores a los mismos y después manipularlos.

2. COMPETENCIA

Se logra integrar conocimientos previos para trabajar con arreglos de manera correcta.
Se entienden las diferencias entre arreglo unidimensional, bidimensional y tridimensional.
Se manipulan los arreglos de forma correcta y consciente.

3. FUNDAMENTOS

Cuando declaramos una variable estamos apartando en memoria espacio para guardar sus posibles valores dependiendo del tipo de dato que se trata. Un array o arreglo son una serie de localidades en memoria consecutivas que están asignadas a un solo nombre y un mismo tipo de datos.

Mas información aqui:

<https://drive.google.com/drive/folders/1y0XlG-5yPiklleEGsQoPXNsuTMz7xkOE>

4. PROCEDIMIENTO

- 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



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

5. RESULTADOS Y CONCLUSIONES

Los arreglos nos ayudan a guardar gran cantidad de datos de un mismo tipo utilizando una sola variable, lo que nos ayuda a guardar memoria. Además estos son útiles cuando queremos guardar gran cantidad de datos y después manipular los mismos, ya sea hacer cálculos con ellos, compararlos entre sí, ordenarlos, etc.}

```
void LlenarMatriz(int matriz[][4], int vect1[], int vect2[], int m, int n)
{
    // VARIABLES LOCALES
    int i, j, k;
    system("CLS");
    // AQUI DESARROLLO PROGRAMA
    for (i = 0, k=0; i < m; i++)
    {
        for (j = 0; j < n; j++, k++)
        {
            if (k < 10) //Llena la matriz con los 10 datos del vector 1
            {
                matriz[i][j] = vect1[k];
            }
            else //Cuando se termina el vector 1 continua con el vector 2
            {
                matriz[i][j] = vect2[k-10];
            }
        }
    }
    printf("Matriz rellena\n");
    system("PAUSE");
}
```

6. ANEXOS

Archivo anexo.



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