



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: López Michel José Alonso

Matrícula: 370650

Maestro: Pedro Núñez Yépiz

Actividad No. 8: Arreglos en C.

Tema - Unidad: Arreglos y funciones – Unidad 1.



Ensenada Baja California a 30 de 09 del 2023



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. INTRODUCCIÓN

El alumno utilizara lo enseñado en clase por medio de los arreglos, y también de la mano de las funciones, para así de esta manera poder lograr hacer matrices en base a los conocimientos previos de vectores con cadenas.

2. COMPETENCIA

El alumno llevará a cabo los ejercicios planteados con ayuda del profesor que dejo en un documento e implementará los arreglos en los ejercicios, además que utilizará funciones para poder hacer un poco más práctico y entendible su programa.

3. FUNDAMENTOS

El alumno utilizara lo enseñado en clase para dar practica a los arreglos y comprenda los fundamentos de este. También, si el alumno queda con dudas respecto a la actividad, tiene como herramientas de apoyo de internet, del profesor o de alumnos.

4. PROCEDIMIENTO

ACTIVIDAD 8

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 Y CONCLUSIONES

LMJ_A08_01_432

```
// Lopez Michel Jose Alonso   Matricula: 370650
// Fecha inicio: 27/09/2023   Fecha fin: 30/09/2023

// En este programa el usuario debera ingresar solo diez datos para llenar el vector uno y llenara la matriz, en cambio
// la computadora aleatoriamente llenara el vector dos y tambien lo hara en base al vector 1 y 2 para despues emprimirlos
//, tambien cuando el usuario llenara la matriz la computadora imprimira la matriz en base a el vector 1 y2.
// LMJA_A08_01_432

//-----Librerias-----
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#define M 10

//-----
// ./semana8/LMJ_A08_01_432

//*****
void menu();
void LlenarVector1_Manualmente (int vect1[],int m);
void LlenarVector2_Aleatoriamente (int vect2[],int m);
bool Sin_repetir ();
void LlenarVector3_ConVector1_Y_VECTOR2 (int vect1[],int vect2[],int vect3[]);
void Imprimir_Vectores (int vect1[],int vect2[],int vect3[]);
void LlenaMatriz_4X4 (int vect1[],int vect2[],int vect3[],int m[][4]);
void Imprimir_Matriz (int m[][4]);
//*****

//***** Main principal *****
int main()
{
    menu();
    return 0;
}
//*****

//*****FUNCIONES*****
//*****El menu donde mandaremos a llamar a todas las demas funciones*****

void menu()
{
    int op;
    int vect1[M], vect2[M], vect3[20]; // Declaración de vectores aquí
    int m[4][4]; // Declaración de la matriz aquí
    do {
        system("CLS");
        printf(" M E N U \n");
        printf("1.- LLENAR VECTOR 1 (MANUALMENTE)\n");
        printf("2.- LLENAR VECTOR 2 ALEATORIAMENTE\n");
        printf("3.- LLENAR VECTOR 3 (CON VECTOR1 Y VECTOR2)\n");
        printf("4.- IMPRIMIR VECTORES\n");
        printf("5.- LLENA MATRIZ 4 X 4\n");
        printf("6.- IMPRIMIR MATRIZ\n");
        printf("0.- SALIR\n");
        printf("ESCOGE UNA OPCION: ");
        scanf("%d", &op);

        switch (op) {
            case 1:
                LlenarVector1_Manualmente(vect1, M);
                break;

            case 2:
                LlenarVector2_Aleatoriamente(vect2, M);
                break;

            case 3:
                LlenarVector3_ConVector1_Y_VECTOR2(vect1, vect2, vect3);
                break;

            case 4:
                Imprimir_Vectores(vect1, vect2, vect3);
                break;

            case 5:
                LlenaMatriz_4X4 (vect1,vect2,vect3,m);
                break;

            case 6:
                Imprimir_Matriz (m);
                break;

            case 0:
                printf("Saliendo del programa.\n");
                break;

            default:
                printf("Opción no válida. Intente de nuevo.\n");
                break;
        }
    } while (op != 0);
}
//*****
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
PS C:\Users\Usuario\Desktop\programaspe> ./semana8/LM0_A08_01_432
```

M E N U

- 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
- ESCOGE UNA OPCION:

M E N U

- 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
- ESCOGE UNA OPCION: 0
Saliendo del programa.
PS C:\Users\Usuario\Desktop\programaspe>

Conclusión:

En esta práctica por medio de los arreglos y matrices pudimos lograr entender el funcionamiento de los mismo, haciendo énfasis en que para poder hacer todo esto primero que nada utilizamos las funciones, en esta practica nos alejamos del tema de cadena, pero yendo al tema de los arreglos un poco mas afondo he de decir que por el lado de los unidimensionales solo se utiliza el valor que estará en los corchetes como m, que es la fila y n los bidimensionales utilizaremos m como filas y n como columnas, estas dos mismas fusionándose para así crear una matriz con la cual podemos manipular los datos a voluntad según sea el caso. Claro que también existen los multidimensionales, pero para ejemplos prácticos solo usaremos los dos anteriores, pero sobre todo los bidimensionales, también recordemos que usaremos vectores en forma de usar dos variables como i y j. Ahora también recordemos que para declarar un vector tiene que tener el tipo de dato como int, por ejemplo, el nombre que ese cada quien decide cual ponerle y el tamaño o la cantidad que tendrá dicho arreglo, por supuesto que también podemos declarar un vector con valor como constante o variable.

6. 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 Sznajder, 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



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño