Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave **36276**

Alumno: Antonio Ramos González

Matrícula: 372576

Maestro: Pedro Núñez Yépiz

Actividad No. 9: FUNCIONES y METODOS DE ORDENACION Y BUSQUEDA

Tema - Unidad 5 : Arreglos y Cadenas

Ensenada Baja California a 5 de octubre del 2023

1. INTRODUCCIÓN

Los arreglos son una estructura fundamental que permite el almacenamiento de distintas colecciones de datos del mismo tipo bajo un mismo nombre, la almacenacion de dichos datos dentro de un vector facilita enormemente su acceso y manipulación, por lo que son sumamente útiles cuando se trabaja con una gran cantidad de datos.

Las librerías son elementos esenciales que todo programa debe de llevar. Es por esa razón que distintos desarrolladores buscan crear su propia librería, la cual contiene distintas funciones que suelen usar a lo largo de su programa y todas ellas con un propósito general.

2. COMPETENCIA

La competencia que se busca para nosotros los alumnos es Fortalecer los conocimientos comprendidos en prácticas anteriores, como lo son las estructuras de control, los distintos tipos de ciclos y sus aplicaciones, así como los métodos de validación y las distintas funciones.

Lo que se busca es seguir fortaleciendo los conocimientos sobre los arreglos y cadenas, agregando distintos métodos de ordenación y búsqueda, así como iniciar con la creación de nuestra biblioteca, en la cual se encontraran las funciones usadas en prácticas anteriores y se agregaran las futuras funciones creadas.

3. FUNDAMENTOS

Los arreglos son colecciones de variables del mismo tipo donde se guarda todo tipo de información, a la cual se accede mediante un índice, el cual indica en donde se guarda dicha información, siendo la dirección más baja el primer elemento y la más ala el último elemento.

Existen diferentes dimensiones que puede tener un arreglo, siendo el arreglo unidimensional el más básico de todos, el cual se asigna de la siguiente manera

Int vector[tamaño]

Los elementos de los arreglos en direcciones controladas por el tamaño, dichas direcciones iniciaran en 0 y terminaran en n-1, por ejemplo

Int vect[10](ira del 0 al 9)

La asignación de un valor al arreglo se presenta de la siguiente forma

Int vect[10]

I=5

Vect [i]= 255 (i representando la dirección del índice en donde se guardará la información)

Los arreglos bidimensionales, también conocidos como matrices son un tipo de arreglo que almacena información guardada en diferentes filas y columnas.

Int matriz [filas][columnas]

La manera en que las matrices leen sus datos es de izquierda a derecha, leyendo primero las fila y posteriormente las columnas, cuando llega a la última columna regresa a la columna 1 pero ahora se encuentra en la siguiente fila

Int mtz[4][4]

I=65

Mtz[1][3]=i(asignando el valor de i en la fila 1 columna 3)

Nuñez,Yepiz,P.(2022)Programación estructurada

4. PROCEDIMIENTO

MENÚ
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0.- SALIR

NOTA: El programa deberá repetirse cuantas veces lo desee el usuario, Validado el menú con la función vali_num

I N S T R U C C I O N E S

1.- LLENAR VECTOR .- Llenar vector con 15 números, los números generados aleatoriamente, los números entre el rango de 100 al 200 **(no repetidos)**
2.- LLENAR MATRIZ .- Llenar la matriz de 4x4 con con números generados aleatoriamente, números entre el rango de 1 al 16 **(no repetidos)**
3.- IMPRIMIR VECTOR .- Imprime el vector que se envíe, donde la función recibe como parámetro el vector,tamaño, nombre del vector.
4.- IMPRIMIR MATRIZ.- Imprime la matriz sin importar el tamaño de la matriz recibiendo como parámetros la matriz, la cantidad de renglones y columnas, así como nombre que se le dará a la matriz
5.- ORDENAR VECTOR.- Usar función que ordene el vector por el método de ordenación de la **Burbuja mejorada**.
6.- BUSCAR VALOR EN VECTOR.- Buscar un valor en el vector usando el método de **búsqueda secuencial**.
0.- SALIR

5. RESULTADOS Y CONCLUSIONES

En conclusión, el uso de vectores y las matrices para el almacenamiento de información son estructuras fundamentales, con las cuales podemos trabajar el manejo de los datos de una manera más eficiente. También la creación de nuestras propias bibliotecas para almacenar nuestras funciones es realmente importante, ahorrándonos el tiempo de escribir dicho bloque de código o buscarlo para copearlo en nuestros anteriores programas, ya que solo necesitamos llamarlas y estas funcionaran de manera correcta, así como poder actualizarla cuando usemos nuevas funciones

Avtividad_9

```c
int msg()
{
    short int opc;//guardara la decision del usuario
    system("CLS");
    printf("MENU\n");
    printf("1.- LLENA VECTOR 1(MANUALMENTE)\n");
    printf("2.- LLENA VECTOR 2(ALEATORIAMENTE)\n");
    printf("3.- LLENA VECTOR 3(VECTOR 1 Y VECTOR 2)\n");
    printf("4.- IMPRIMIR VECTOES\n");
    printf("5.- LLENA MATRIZ(VECTOR 1 Y VECTOR 2)\n");
    printf("6.- IMPRIMIR MATRIZ\n");
    printf("0.- SALIR\n");
    printf("ESCOGE UNA OPCION: ");
    scanf("%hd", &opc);//lee la decicion del usuario
    return opc;//retorna opc
}

void menu()
{
    int vect1[10], vect2[10], vect3[20];//definimos el tamaño de los vetores
    int mtz[4][4];//definimos el tamaño de la matriz
    short int opc;
    do
    {
        opc = msg();//leemos el valor retornado de la funcion msg
        switch (opc)
        {
        case 1:
            vect_1(vect1, 10);//se manda vect1, asi como su tamaño a la funcion

            break;
        case 2:
            vect_2(vect2, 1, 20);//se manda vect2, asi como su limite de datos a la funcion
            break;
        case 3:
            vect_3(vect1, vect2, vect3, 20);//se manda vect1,vect2 para llenar vect3
            break;
        case 4:
            imprimir_vetores(vect1, vect2, vect3, 10, 20);//se manda vect1,vect2 y vect3 para imprimirlos
            break;
        case 5:
            matriz(vect1, vect2, mtz, 4, 10);//se manda vect1,vect2 para llenar mtz
            break;
        case 6:
            imprimir_matriz(mtz,4);//se manda mtz para imprimirla
            break;
        default:
            break;
        }

    } while (opc != 0);
}
```

Biblioteca

```c
// Antonio Ramos Gonzalez Mt: 372576
// 3/10/2023 || 10/5/2023
// Esta biblioteca contiene las funciones que usare durante proximas practicas
// Babilonia
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<string.h>

int valid (const char msg[],int lim_inf,int lim_sup);
void vectRand(int vect[],int m,int lim_inf,int lim_sup);
void PrintVect(int vect[],int m);
void OrderVect(int vect[],int m);
int SearchVect(int vect[],int m, int num);

//Genera numero aleatorios no repetidos en un vector
void vectRand(int vect[],int m, int lim_in, int lim_sup)···

//Imprime vector
void PrintVect(int vect[],int m)···

//valida las opciones
int valid(const char msg[], int lim_in, int lim_sup)···

//ordena un vector
void OrderVect(int vect[],int m)···

//Busca en el vector
int SearchVect(int vect[],int m,int num)···
```

6. ANEXOS

Link GitHub:https: https://github.com/Anrago/Programacion-estructurada/tree/main/Actividad_9

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