

Ingeniero en computación
Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave **36276**

Alumno: Hernandez Ceseña Ivan Fernando

Matrícula: 373077

Maestro: Pedro Núñez Yépiz

Actividad No. : 8

Tema - Unidad : vectores y matrices

Ensenada Baja California a 01 de octubre del 2023

1. INTRODUCCIÓN

Se van a usar funciones para tener un programa con un operador múltiple (switch) y poder acceder por medio de este a la acción que se quiere ejecutar. acompañados de ciclos para automatizar selecciones y entradas de datos. Utilizando vectores y matrices.

2. COMPETENCIA

usar funciones para hacer un código más claro fácil de entender, además de poner en práctica la optimización de código. utilizando vectores y matrices.

3. FUNDAMENTOS

- sintaxis de ciclos.
- funcionamiento de ciclos.
- proceso de ciclos.
- movilización de una función a otra.
- funcionamiento de vectores y matrices y cómo utilizarlas correctamente.
- declaraciones de vectores y matrices.

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 ARREGLO 1 Y ARREGLO 2

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

5. RESULTADOS Y CONCLUSIONES

De esta manera se declaran los prototipos de todas las funciones que vamos a usar en el programa. Tomando en cuenta todos los vectores y/o matrices que vamos a utilizar en el programa.

```
void menu(void);
int msg(void);
int reiniciar(void);
int validar(char [], int li, int ls);

void vectorManual(int vector1[], int n);
void vectorAleatorio(int vector2[], int n);
void vectores(int vector1[], int vector2[], int vector3[], int n);
void imprimirVectores(int *vector1, int n);
void llenarMatriz(int *vector1, int *vector2, int matriz[][4]);
void imprimirMatriz(int matriz[][4], int m);
```

esta es la manera de declarar la funcion en la que vas a trabajar. (son iguales las declaraciones)

```
void vectorManual(int vector1[], int n)
{
    int num;
    for(int i=0; i<n; i++)
    {
        // printf( "dame un numero entre 30 y 70: \n");
        vector1[i]= num=validar("dame un numero entre 30 y 70\n", 30, 70);
    }
}

void vectorAleatorio(int vector2[], int n)
{
    srand(time(NULL));
    for(int i=0; i<n; i++)
    {
        vector2[i]= (rand())% 20)+1;
    }
}

void vectores(int vector1[], int vector2[], int vector3[], int n)
{
    for(int i=0; i<n; i++)
    {
        vector3[i]= vector1[i];
        vector3[i+n]= vector2[i];
    }
}

void imprimirVectores(int *vector, int n)
{
    for(int i=0; i<n; i++)
    {
        printf("%d ", vector[i]);
    }
    printf("\n");
}

void llenarMatriz(int vector1[], int vector2[], int matriz[][4])
{
    int pos=0, i, j;
    for( i=0; i<4; i++)
    {
        for( j=0; j<4; j++)
        {
            if(pos<10)
            {
                matriz[i][j]= vector1[pos];
            }
            else
            {
                matriz[i][j]= vector2[pos-10];
            }
            // printf("%d ", matriz[i][j]);
            pos++;
        }
    }
}

void imprimirMatriz(int matriz[][4], int m)
{
    for(int i=0; i<m; i++)
    {
        for(int j=0; j<4; j++)
        {
            printf("[%d] ", matriz[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}
```

y esta es la manera en la que se mandan parametros al invocar una funcion que involucre vectores y matrices.

```
void menu()
{
    int op;
    int n=10;
    int m=4;
    int vector1[n];
    int vector2[n];
    int vector3[n+n];
    int matriz[m][m];
    // n=10;
    do
    {
        // fflush(stdin);
        // gets(op);
        // int vector1[n];
        // int vector2[n];
        // int vector3[n+n];
        // int matriz[3][3];
        fflush(stdin);
        op= msg();
        switch(op)
        {
            // fflush(stdin);
            // op= msg();
            case 1:
                vectorManual(vector1, n);
                break;

            case 2:
                vectorAleatorio(vector2, n);
                break;

            case 3:
                vectores(vector1, vector2, vector3, n);
                break;

            case 4:
                printf("vector 1 (llenado manualmente)\n");
                imprimirVectores(vector1, 10);
                printf("vector 2 llenado aleatoriamente\n");
                imprimirVectores(vector2, 10);
                printf("vector 3 (llenado a base de los vectores 1 y 2)\n");
                imprimirVectores(vector3, 20);
                break;

            case 5:
                llenarMatriz(vector1, vector2, matriz);
                break;

            case 6:
                imprimirMatriz(matriz, m);
                break;
        }
    } while (op != 0);
}
```

6. ANEXOS

anexos en el otro archivo:

nombre del archivo: anexo_HCIF_RP08_PE

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