

SEMANA #11 CONTINUACION DE FUNCIONES.

```
/* Realice una función que tome como parámetros un vector de números enteros
y devuelva la suma de sus elementos.*/

#include<iostream>
#include<stdlib.h>
using namespace std;

void pedirDatos();
int calcularSuma(int vec[], int);

int vec[100], tam;

int main() {
    pedirDatos();

    cout << "\nLa suma es: " << calcularSuma(vec, tam) << endl;

    system("pause");
    return 0;
}

void pedirDatos() {
    cout << "Digite el numero de elementos del arreglo: ";
    cin >> tam;

    for (int i = 0; i < tam; i++) {
        cout << i + 1 << ". Digite un numero: ";
        cin >> vec[i];
    }
}

int calcularSuma(int vec[], int tam) {
    int suma = 0;

    for (int i = 0; i < tam; i++) {
        suma += vec[i];
    }

    return suma;
}
```

```
/*Realiza una función que tome como parámetros un vector de números y su
tamaño y cambie el signo de los elementos del vector.*/

#include<iostream>
#include<stdlib.h>
using namespace std;

//Prototipos de Funciones
void pedirDatos();
void cambiarSigno(int vec[], int);
void mostrarVector(int vec[], int);

int tam, vec[100];

int main() {
    //Llamada a las funciones - en orden
    pedirDatos();
    cambiarSigno(vec, tam);
    mostrarVector(vec, tam);

    system("pause");
    return 0;
}

void pedirDatos() {
    cout << "Digite el tamaño del vector: ";
    cin >> tam;

    for (int i = 0; i < tam; i++) {
        cout << i + 1 << ". Digite un numero: ";
        cin >> vec[i];
    }
}

void cambiarSigno(int vec[], int tam) {
    for (int i = 0; i < tam; i++) {
        vec[i] *= -1;
    }
}

void mostrarVector(int vec[], int tam) {
    cout << "\nMostrando vector con cambio de signo: " << endl;
    for (int i = 0; i < tam; i++) {
        cout << vec[i] << " ";
    }
}
```

```
/*Escriba una función nombrada tiempo() que tenga un parámetro en número
entero llamado totalSeg y tres parámetros de referencia enteros nombrados horas, min
y seg. La función es convertir el número de segundos transmitido en un número
equivalente de horas, minutos y segundos.*/

#include<iostream>
#include<stdlib.h>
using namespace std;

void tiempo(int, int&, int&, int&);

int main() {
    int totalSeg, horas, min, seg;

    cout << "Digita la cantidad de segundos: ";
    cin >> totalSeg;

    tiempo(totalSeg, horas, min, seg);

    cout << "\nHoras: " << horas << endl;
    cout << "Minutos: " << min << endl;
    cout << "Segundos: " << seg << endl;

    system("pause");
    return 0;
}

void tiempo(int totalSeg, int& horas, int& min, int& seg) {
    horas = totalSeg / 3600;
    totalSeg %= 3600;
    min = totalSeg / 60;
    seg = totalSeg % 60;
}
```

```
/*Desarrollar una función que determine si una matriz es simétrica o no.

Una matriz es simetrica:  $A = A^t$ 

- La matriz debe ser cuadrada. _--
-  $A_{ij} = A_{ji}$ 

|2  5  9|      |2  5  9|
|5  8 -1|  ->  |5  8 -1|
|9 -1 10|      |9 -1 10|

*/

#include<iostream>
#include<stdlib.h>
using namespace std;

int main() {
    pedirDatos();
    comprobarSimetria(m, nfilas, ncol);

    system("pause");
    return 0;
}

void pedirDatos() {
    cout << "Digite el numero de filas: ";
    cin >> nfilas;
    cout << "Digite el numero de columnas: ";
    cin >> ncol;

    //Pedimos todos los elementos de la matriz
    for (int i = 0; i < nfilas; i++) {
        for (int j = 0; j < ncol; j++) {
            cout << "Digite m[" << i << "][" << j << "]: ";
            cin >> m[i][j];
        }
    }
}
```

```
void comprobarSimetria(int m[][100], int nfilas, int ncol) {
    int cont = 0;

    //Si es cuadrada
    if (nfilas == ncol) {
        //y ademas si Aij = Aji
        for (int i = 0; i < nfilas; i++) {
            for (int j = 0; j < ncol; j++) {
                if (m[i][j] == m[j][i]) {
                    cont++; //El conteo debe ser igual al numero de elementos de la matriz
                }
            }
        }

        if (cont == nfilas * ncol) {
            cout << "\nLa matriz es Simetrica";
        }
        else {
            cout << "\nLa matriz NO es simetrica";
        }
    }
}
```