

Semana #10 Continuación de Estructuras.

```
/*Hacer 2 estructuras una llamada promedio que tendrá los siguientes campos:
nota1, nota2, nota3; y otro llamada alumno que tendrá los siguientes miembros:
nombre, sexo, edad; hacer que la estructura promedio este anidada en la
estructura alumno, luego pedir todos los datos para un alumno, luego calcular su
promedio, y por ultimo imprimir todos sus datos incluidos el promedio.*/

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

struct Promedio {
    float nota1;
    float nota2;
    float nota3;
};

struct Alumno {
    char nombre[20];
    char sexo[10];
    int edad;
    struct Promedio prom;
}alumno1;

int main() {
    float promedio_alumno;

    cout << "Nombre: "; cin.getline(alumno1.nombre, 20, '\n');
    cout << "Sexo: "; cin.getline(alumno1.sexo, 10, '\n');
    cout << "Edad: "; cin >> alumno1.edad;

    cout << ".:Notas del Examen:." << endl;
    cout << "Nota1: "; cin >> alumno1.prom.nota1;
    cout << "Nota2: "; cin >> alumno1.prom.nota2;
    cout << "Nota3: "; cin >> alumno1.prom.nota3;

    //Sacando el promedio del alumno
    promedio_alumno = (alumno1.prom.nota1 + alumno1.prom.nota2 + alumno1.prom.nota3) / 3;

    cout << "\n\nMostrando Datos" << endl;
    cout << "Nombre: " << alumno1.nombre << endl;
    cout << "Sexo: " << alumno1.sexo << endl;
    cout << "Edad: " << alumno1.edad << endl;
    cout << "Promedio: " << promedio_alumno << endl;

    system("pause");
    return 0;
}
```

```
/* Hacer una estructura llamada corredor, en la cual se tendrán los siguientes
campos : Nombre, edad, sexo, club, pedir datos al usuario para un corredor, y asignarle
una categoría de competición :
-Juvenil <= 18 años
- Señor <= 40 años
- Veterano > 40 años
Posteriormente imprimir todos los datos del corredor, incluida su categoría de
competición.*/

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

struct Corredor {
    char nombre[20];
    int edad;
    char sexo[10];
    char club[20];
}c1;

int main() {
    char categoria[20];

    cout << "Nombre: "; cin.getline(c1.nombre, 20, '\n');
    cout << "Edad: "; cin >> c1.edad;
    while (getchar() != '\n');
    cout << "Sexo: "; cin.getline(c1.sexo, 10, '\n');
    cout << "Club: "; cin.getline(c1.club, 20, '\n');

    if (c1.edad <= 18) {
        strcpy_s(categoria, "Juvenil");
    }
    else if (c1.edad <= 40) {
        strcpy_s(categoria, "Senior");
    }
    else {
        strcpy_s(categoria, "Veterano");
    }

    cout << "\n\nMostrando Datos" << endl;
    cout << "Nombre: " << c1.nombre << endl;
    cout << "Edad: " << c1.edad << endl;
    cout << "Sexo: " << c1.sexo << endl;
    cout << "Club: " << c1.club << endl;
    cout << "Categoría: " << categoria << endl;
    system("pause");
    return 0;
}
```

```
/*8. Defina una estructura que sirva para representar a una persona. La estructura
debe contener dos campos: el nombre de la persona y un valor de tipo lógico que indica
si la persona tiene algún tipo de discapacidad. Realice un programa que dado un vector
de personas rellene dos nuevos vectores: uno que contenga las personas que no tienen
ninguna discapacidad y otro que contenga las personas con discapacidad.*/

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

struct Persona_discapacidad {
    char nombre[30];
    bool discapacidad;
}personas[30], personasConD[30], personasSinD[30];
//personasConD = Persona con Discapacidad
//personasSinD = Persona sin Discapacidad

int main() {
    int n_personas, j = 0, k = 0;

    cout << "Digite el numero de personas: ";
    cin >> n_personas;
    //Rellenar el arreglo de estructuras
    for (int i = 0; i < n_personas; i++) {
        while (getchar() != '\n');

        cout << "Nombre : "; cin.getline(personas[i].nombre, 30, '\n');
        cout << "Discapacidad (1/0)? :"; cin >> personas[i].discapacidad;

        //Almacenamos las personas con discapacidad
        if (personas[i].discapacidad == 1) {
            strcpy_s(personasConD[j].nombre, personas[i].nombre);
            j++; //Es un nuevo arreglo por lo tanto debe iniciar en 0 y aumentar
        }
        //Almacenamos las personas sin Discapacidad
        else {
            strcpy_s(personasSinD[k].nombre, personas[i].nombre);
            k++; //Es un nuevo arreglo por lo tanto debe iniciar en 0 y aumentar
        }
        cout << "\n";
    }

    //Mostrar las personas con Discapacidad
    cout << "\nPersonas con Discapacidad\n";
    for (int i = 0; i < n_personas; i++) {
        cout << personasConD[i].nombre << endl;
    }
    //Mostrar las personas sin Discapacidad
    cout << "\nPersonas sin Discapacidad\n";
    for (int i = 0; i < n_personas; i++) {
        cout << personasSinD[i].nombre << endl;
    }

    system("pause");
    return 0;
}
```