



PROGRAMACIÓN

Unidad 7 – Parte 2. Estructuras y funciones. Arreglos de estructuras.

2022

Lic. Mariela A. Velázquez

Repaso de lo visto



Usar campos de una estructura como argumentos de funciones

- ❑ Se trabajara con el campo de la estructura usando el operador punto.
- ❑ Cuando se pasa un campo o miembro de una estructura a una función como argumento, en realidad se pasa el valor de ese campo.

OPERACIONES



Usar variables estructuradas por partes como argumentos de funciones:

- Al pasar un campo de la estructura a una función, se puede usar el mecanismo de pasaje por valor o por referencia.

Ejemplo

Supongamos que tenemos una estructura de nombre **Alumno** que almacena los datos de un alumno como ser sus datos personales (apellido, nombre, edad) y las notas trimestrales del mismo. Nuestra tarea será realizar una función que calcule el promedio de 3 notas.

```
struct{  
    char nombre[100];  
    char apellido[100];  
    int edad;  
    float nota1;  
    float nota2;  
    float nota3;  
}typedef Alumno;
```

```
float promedio(float nota1, float nota2, float nota3);  
int main()  
{  
    Alumno estudiante;  
    float prom;  
  
    printf("Ingrese el Apellido del alumno: ");  
    gets(estudiante.apellido);  
  
    printf("\nIngrese el Nombre del alumno: ");  
    gets(estudiante.nombre);  
  
    printf("\nIngrese la edad del alumno: ");  
    scanf("%d", &estudiante.edad);  
  
    printf("\nIngrese la 1ra nota:");  
    scanf("%f", &estudiante.nota1);  
  
    printf("\nIngrese la 2da nota:");  
    scanf("%f", &estudiante.nota2);  
  
    printf("\nIngrese la 3ra nota:");  
    scanf("%f", &estudiante.nota3);  
  
    prom = promedio(estudiante.nota1, estudiante.nota2, estudiante.nota3);  
  
    printf("\nEl promedio del almno es: %f", prom);  
  
    return 0;  
}  
  
float promedio(float nota1, float nota2, float nota3)  
{  
    return ((nota1 + nota2 + nota3) / 3);  
}
```

Ejemplo

Podemos modificar la nota del alumno según la consideración que se le de en la materia.

```
struct{  
    char nombre[100];  
    char apellido[100];  
    int edad;  
    float nota1;  
    float nota2;  
    float nota3;  
}typedef Alumno;
```

```
void cambiarNotas(float *pnota1, float *pnota2, float *pnota3);  
int main()  
{  
    Alumno estudiante;  
    float prom;  
  
    printf("Ingrese el Apellido del alumno: ");  
    gets(estudiante.apellido);  
  
    printf("\nIngrese el Nombre del alumno: ");  
    gets(estudiante.nombre);  
  
    printf("\nIngrese la edad del alumno: ");  
    scanf("%d", &estudiante.edad);  
  
    printf("\nIngrese la 1ra nota:");  
    scanf("%f", &estudiante.nota1);  
  
    printf("\nIngrese la 2da nota:");  
    scanf("%f", &estudiante.nota2);  
  
    printf("\nIngrese la 3ra nota:");  
    scanf("%f", &estudiante.nota3);  
  
    cambiarNotas(&estudiante.nota1, &estudiante.nota2, &estudiante.nota3);  
    prom = promedio(estudiante.nota1, estudiante.nota2, estudiante.nota3);  
  
    printf("\nEl promedio del alumno es: %f", prom);  
  
    return 0;  
}  
  
void cambiarNotas(float *pnota1, float *pnota2, float *pnota3)  
{  
    *pnota1 = (*pnota1) * 0.2;  
    *pnota2 = (*pnota2) * 0.2;  
    *pnota3 = (*pnota3) * 0.6;  
}
```

OPERACIONES

Usar variables estructuradas completas como argumentos de funciones:

- Al pasar una estructura completa a una función, se usa el mecanismo de pasaje por valor.
- Sabemos que de esta manera cualquier cambio en los contenidos de los campos de la estructura dentro de la función, no se reflejan en el bloque que la invoco.

Ejemplo

Continuemos con el ejemplo anterior

```
struct{  
    char nombre[100];  
    char apellido[100];  
    int edad;  
    float nota1;  
    float nota2;  
    float nota3;  
}typedef Alumno;
```

```
void mostrarDatos(Alumno alum)  
{  
    puts("Los datos cargado del alumno es");  
    printf("El Apellido del alumno es: ");  
    puts(alum.apellido);  
  
    printf("\nEl Nombre del alumno es: ");  
    puts(alum.nombre);  
  
    printf("\nLa edad del alumno es: %d", alum.edad);  
  
    printf("\nLa 1ra nota es: %d", alum.nota1);  
  
    printf("\nLa 2da nota es: %d", alum.nota2);  
  
    printf("\nLa 3ra nota es: %d", alum.nota3);  
}
```

```
void mostrarDatos(Alumno alum);  
int main()  
{  
    Alumno estudiante;  
    float prom;  
  
    printf("Ingrese el Apellido del alumno: ");  
    gets(estudiante.apellido);  
  
    printf("\nIngrese el Nombre del alumno: ");  
    gets(estudiante.nombre);  
  
    printf("\nIngrese la edad del alumno: ");  
    scanf("%d", &estudiante.edad);  
  
    printf("\nIngrese la 1ra nota:");  
    scanf("%f", &estudiante.nota1);  
  
    printf("\nIngrese la 2da nota:");  
    scanf("%f", &estudiante.nota2);  
  
    printf("\nIngrese la 3ra nota:");  
    scanf("%f", &estudiante.nota3);  
  
    mostrarDatos(estudiante);  
    return 0;  
}
```


Ejemplo

Mejoremos el ejemplo anterior

```
struct{
    char nombre[100];
    char apellido[100];
    int edad;
    float nota1;
    float nota2;
    float nota3;
}typedef Alumno;
```

```
void mostrarDatos(Alumno alum)
{
    puts("Los datos cargado del alumno es");
    printf("El Apellido del alumno es: ");
    puts(alum.apellido);

    printf("\nEl Nombre del alumno es: ");
    puts(alum.nombre);

    printf("\nLa edad del alumno es: %d", alum.edad);

    printf("\nLa 1ra nota es: %d", alum.nota1);

    printf("\nLa 2da nota es: %d", alum.nota2);

    printf("\nLa 3ra nota es: %d", alum.nota3);
}
```

```
Alumno cargarDatos();
int main()
{
    Alumno estudiante;

    estudiante = cargarDatos();

    mostrarDatos(estudiante);
    return 0;
}

Alumno cargarDatos()
{
    Alumno alum;

    printf("Ingrese el Apellido del alumno: ");
    gets(alum.apellido);

    printf("\nIngrese el Nombre del alumno: ");
    gets(alum.nombre);

    printf("\nIngrese la edad del alumno: ");
    scanf("%d", &alum.edad);

    printf("\nIngrese la 1ra nota:");
    scanf("%f", &alum.nota1);

    printf("\nIngrese la 2da nota:");
    scanf("%f", &alum.nota2);

    printf("\nIngrese la 3ra nota:");
    scanf("%f", &alum.nota3);

    return (alum);
}
```

Arreglo de estructura

- Ahora necesitamos sacar el promedio de notas de un grupo de 20 alumnos.
- ¿Qué deberíamos hacer para responder a ese requisito?
- Tengo que declarar 20 variables de tipo

struct?

No



Arreglo de estructura

- ¿Cómo sería un arreglo con todos los datos de un alumno?

Alumnos

Nombre	Nombre	Nombre		Nombre
Apellido	Apellido	Apellido		Apellido
Edad	Edad	Edad		Edad
Nota1	Nota1	Nota1		Nota1
Nota2	Nota2	Nota2		Nota2
Nota3	Nota3	Nota3		Nota3
[0]	[1]	[2]	...	[n]



Arreglo de estructura

- Respondemos esas preguntas desarrollando el siguiente ejemplo:

```
struct Alumno{  
    char nom[100];  
    char ape[100];  
    int edad;  
    float nota1;  
    float nota2;  
    float nota3;  
};
```

En este ejemplo podemos ver que trabajamos con arreglos de estructuras, haciendo uso del índice, como lo veníamos haciendo con los arreglos en general.

```
float promedio(float nota1, float nota2, float nota3);  
  
int main()  
{  
    struct Alumno estudiante[20];  
    float prom;  
  
    for (int i = 0; i < 20; i++)  
    {  
        printf("Datos del alumno N°: %d", i);  
  
        puts("Ingrese el apellido del alumno : ");  
        gets(estudiante[i].ape);  
  
        puts("Ingrese el nombre del alumno:");  
        gets(estudiante[i].nom);  
  
        puts("Ingrese la edad del alum:");  
        scanf("%d", &estudiante[i].edad);  
  
        puts("Ingrese 1ra nota:");  
        scanf("%f", &estudiante[i].nota1);  
  
        puts("Ingrese 2da nota:");  
        scanf("%f", &estudiante[i].nota2);  
  
        puts("Ingrese 3ra nota:");  
        scanf("%f", &estudiante[i].nota3);  
  
        prom = promedio(estudiante[i].nota1, estudiante[i].nota2, estudiante[i].nota3);  
  
        printf("El promedio del alumno es: %f", prom);  
    }  
  
    return 0;  
}  
  
float promedio(float nota1, float nota2, float nota3)  
{  
    return((nota1+nota2+nota3)/3);  
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hasta la próxima clase!!\n");
```

```
    return 0;
```

```
}
```