



Estructuras



Programación en C.

Arrays vs Estructuras

Arrays

Permiten almacenar un conjunto de datos del mismo tipo.

Ejemplo: podemos almacenar un conjunto de notas, pero no el nombre del alumno.

Estructuras

Admiten variables de distinto tipo.



Estructuras

Una estructura es como una ficha, que permite almacenar distintos campos (variables).

FICHA_ALUMNO

Nombre:

Apellidos:

DNI:

Notas:

Nota Media:



Declaración de Estructuras

Lo primero que hay que hacer es crear la estructura.

Se indicarán los campos que forman la estructura.

Sintaxis

```
struct identificador {  
    tipo identificador_campo1;  
    tipo identificador_campo2;  
    tipo identificador_campoN;  
};
```

Ejemplo

```
struct Tficha{  
    char nombrealumno[30];  
    char apellidos[100];  
    int dni;  
    float notas[5];  
    float nota_media;  
};
```

Declaración de Estructuras

Lo primero que hay que hacer es crear la estructura.

Se indicarán los campos que forman la estructura.

Sintaxis

```
struct identificador {  
    tipo identificador_campo1;  
    tipo identificador_campo2;  
    tipo identificador_campoN;  
};
```

Ejemplo

```
struct Tficha{  
    char nombrealumno[30];  
    char apellidos[100];  
    int dni;  
    float notas[5];  
    float nota_media;  
};
```

Declaración de Estructuras

Lo primero que hay que hacer es crear la estructura.

Se suele declarar fuera del main para que sean accesibles por todo el programa.

```
//Declaración de librerías (include)
//Declaración de constantes (define)
//Declaración de estructuras:
struct Tficha{
    char nombrealumno[30];
    char apellidos[100];
    int dni;
    float notas[5];
    float nota_media;
};
int main (int argc, char *argv[])
```



Declaración de Estructuras

Después hay que declarar la variable de la estructura.

Sintaxis

```
struct identificador nombre_variable;
```



Declaración de Estructuras

Después hay que declarar la variable de la estructura.

Ejemplo

```
//Declaración de librerías (include)
//Declaración de constantes (define)
//Declaración de estructuras:
struct Tficha{
    char nombrealumno[30];
    char apellidos[100];
    int dni;
    float notas[5];
    float nota_media;
};
int main (int argc, char *argv[])
{
    struct Tficha alumno;
```


Declaración de Estructuras

Otra forma de declarar variables de tipo estructura es la siguiente:

Sintaxis

```
struct identificador {  
    tipo identificador_campo1;  
    tipo identificador_campo2;  
    tipo identificador_campoN;  
} nombrevariable;
```

Declaración de Estructuras

Otra forma de declarar variables de tipo estructura es la siguiente:

Ejemplo

```
//Declaración de librerías (include)
//Declaración de constantes (define)
//Declaración de estructuras:
struct Tficha{
    char nombrealumno[30];
    char apellidos[100];
    int dni;
    float notas[5];
    float nota_media;
} alumno;
int main (int argc, char *argv[])
{
    struct Tficha alumno2;
```

Declaración de Estructuras

Podemos crear un array unidimensional de estructuras, por ejemplo, para guardar información de todos los alumnos de la carrera.

FICHA_ALUMNO1

Nombre:
Apellidos:
DNI:
Notas:
Nota Media:

FICHA_ALUMNO2

Nombre:
Apellidos:
DNI:
Notas:
Nota Media:

FICHA_ALUMNO3

Nombre:
Apellidos:
DNI:
Notas:
Nota Media:

Declaración de Estructuras

Podemos crear un array unidimensional de estructuras, por ejemplo, para guardar información de todos los alumnos de la carrera.

Ejemplo

```
//Declaración de librerías (include)
//Declaración de constantes (define)
//Declaración de estructuras:
struct Tficha{
    char nombrealumno[30];
    char apellidos[100];
    int dni;
    float notas[5];
    float nota_media;
};
int main (int argc, char *argv[])
{
    struct Tficha alumnos[100];
```

Acceso a los campos

Podemos acceder a los campos (variables) de una estructura.

Sintaxis

`nombrevariable.identificador_campo`

Ejemplo

`alumno.nota_media`



Acceso a los campos

Es una variable normal, con lo que podemos hacer cualquier operación como con el resto de variables.

```
alumno.nota_media=6.4;  
if (alumno.nota_media < 5)  
...
```

```
scanf ("%f",&alumno.nota_media);  
printf ("%f",alumno.nota_media);  
...
```



Acceso a los campos

Si tenemos un array de estructuras, indicaremos a qué posición del array accedemos y a qué variable en cuestión.

```
alumnos[0].nota_media=6.4;  
if (alumnos[0].nota_media < 5)  
...
```

```
scanf ("%f",&alumnos[0].nota_media);  
printf ("%f",alumnos[0].nota_media);  
...
```



Estructuras. Ejemplo

Lee las notas de 3 alumnos y calcula la nota media

```
#include <stdio.h>

struct Tficha{
    char nombrealumno[30];
    char apellidos[100];
    long dni;
    float notas[5];
    float nota_media;
};

int main (int argc, char *argv[])
{
    int i,j;
    struct Tficha alumnos[3];
    float aux;

    //Leemos los datos
    for (i=0;i<3;i++)
    {
        printf("\nNombre del alumno %d: ",
            (i+1));
        gets(alumnos[i].nombrealumno);
        printf("\nApellidos del alumno %d: ",
            (i+1));
        gets(alumnos[i].apellidos);
        printf("\nDni del alumno %d: ", (i+1));
        scanf("%li",&alumnos[i].dni);
        printf("\nNotas del alumno %d: ",
            (i+1));
        aux=0;
        for (j=0;j<5;j++)
        {
            printf("\nNota %d: ", (j+1));
            scanf("%f",&alumnos[i].notas[j]);
            aux=aux+alumnos[i].notas[j];
        }
        alumnos[i].nota_media=aux/5;
        printf("La nota media del alumno con
            dni %li es %f", alumnos[i].dni,
            alumnos[i].nota_media );
        while(getchar()!='\n');
    }
    return 0;
}
```


Declaración de Estructuras Dinámicas

Otra forma de declarar una estructura es mediante el uso de un puntero. Hay que reservar espacio para la estructura.

Ejemplo

```
//Declaración de librerías (include)
//Declaración de constantes (define)
//Declaración de estructuras:
struct Tficha{
    char nombrealumno[30];
    char apellidos[100];
    int dni;
    float notas[5];
    float nota_media;
};
int main (int argc, char *argv[])
{
    struct Tficha *palumno;
    palumno=(struct Tficha *)malloc (sizeof(struct Tficha));
```

Acceso a los campos.

Podemos acceder a los campos (variables) de una estructura dinámica.

Sintaxis

`nombrepuntero-> identificador_campo |`

Ejemplo

`palumno->nota_media`



Declaración de Estructuras Dinámicas

Otra forma de crear un array unidimensional de estructuras usando punteros, por ejemplo, para guardar información de todos los alumnos de la carrera.

Ejemplo

```
//Declaración de librerías (include)
//Declaración de constantes (define)
//Declaración de estructuras:
struct Tficha{
    char nombrealumno[30];
    char apellidos[100];
    int dni;
    float notas[5];
    float nota_media;
};
int main (int argc, char *argv[])
{
    struct Tficha *palumnos;
    alumnos=(struct Tficha *)malloc (sizeof(struct Tficha)*100);
```

Acceso a los campos.

Podemos acceder a los campos (variables) de un array de estructuras dinámicas.

Sintaxis

`(nombrepuntero+i)-> identificador_campo l`

Ejemplo

`(palumnos+2)->nota_media`



Estructuras dinámicas. Ejemplo

Lee las notas de 3 alumnos y calcula la nota media

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

struct Tficha{
    char nombrealumno[30];
    char apellidos[100];
    long dni;
    float notas[5];
    float nota_media;
};

int main (int argc, char *argv[])
{
    int i,j;
    struct Tficha *palumnos;
    float aux;

    alumnos = (struct Tficha *) malloc(sizeof
        (struct Tficha)*3);
    //Leemos los datos
    for (i=0;i<3;i++)
    {
        printf("\nNombre del alumno %d:",
            (i+1));
        gets((palumnos+i)->nombrealumno);
        printf("\nApellidos del alumno %d:",
            (i+1));
        gets((palumnos+i)->apellidos);
        printf("\nIntroduce el dni del alumno %d:",
            (i+1));
        scanf("%li",&(palumnos+i)->dni);
        printf("\nNotas del alumno %d:", (i+1));
        aux=0;
        for (j=0;j<5;j++)
        {
            printf("\nIntroduzca la nota %d:",
                (j+1));
            scanf("%f",&(palumnos+i)-
                >notas[j]);
            aux=aux+((palumnos+i)->notas[j]);
        }
        (palumnos+i)->nota_media=aux/5;
        printf("La nota media del alumno con dni
            %li es %f", (palumnos+i)->dni,
            (palumnos+i)->nota_media);
    }
    free(palumnos);
    return 0;
}
```