

Unidad 6

ESTRUCTURAS
ESTÁTICAS
ARRAYS

Arrays

DECLARAR, INICIALIZAR Y RECORRER UN ARRAY

Estructuras estáticas - Arrays

Inicialización de arrays unidimensionales

- Se suelen utilizar ciclos FOR aprovechando su índice como índice del array.

```
int miArray[50]; //Declaro un array de 50 posiciones  
  
int i=0; //creo una variable para el índice  
  
for(i=0; i<50;i++){  
    miArray[i]=0; //pongo en 0 todas las posiciones del array  
}
```

En caso de no inicializar el array, tomará los valores que hay en la memoria en ese momento.

Estructuras estáticas - Arrays

Inicialización de arrays multidimensionales

- Se suelen utilizar ciclos FOR aprovechando su índice como índice de la matriz.
- Cada ciclo permite acceder a una posición diferente dentro de la matriz.

```
int miMatriz[10][10];

int d1=0; //declaro una variable para el índice de cada dimensión
int d2=0;

for(d1=0;d1<10;d1++){    //ciclo para la primera dimensión
    for(d2=0;d2<10;d2++){ //ciclo para la segunda dimensión
        miMatriz[d1][d2]=0; //inicialización a 0
    }
}
```

Estructuras estáticas - Arrays

Recorrer un array unidimensional

- Se suelen utilizar ciclos FOR aprovechando su índice como índice del array.
- Cada ciclo permite acceder a una posición diferente dentro del array.

```
int miArray[50]; //Declaro un array de 50 posiciones

int i=0; //creo una variable para el índice

for(i=0; i<50;i++){
    //muestro el contenido de todos los elementos del array
    printf("El valor de la posición %d del array es %d", i, miArray[i]);
}
```

Estructuras estáticas - Arrays

Recorrer un array multidimensional

- Se suelen utilizar ciclos FOR aprovechando su índice como índice del array.
- Se necesita un ciclo por cada dimensión de que tenga.

```
int miMatriz[10][10];

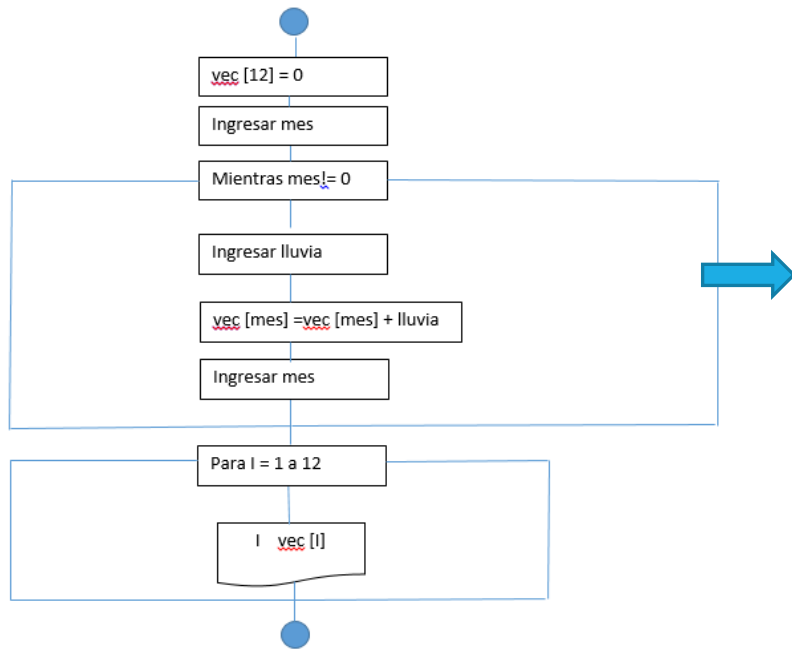
int d1=0; //declaro una variable para el índice de cada dimensión
int d2=0;

for(d1=0;d1<10;d1++){ //ciclo para la primera dimensión
    for(d2=0;d2<10;d2++){ //ciclo para la segunda dimensión
        printf("El valor de la posición %d,%d de la matriz es %d", d1,d2, miMatriz[d1][d2]);
    }
}
```

Estructuras estáticas - Arrays

Carga directa

- Se utiliza el valor ingresado por el usuario como índice del array



```
int vec[12]; //declaro un array para los meses
int mes=0;
int lluvia=0;
printf("ingresar el mes: ");
scanf("%d",&mes);

int i=0;
for(i=0;i<12;i++){
    //inicio el array de meses
    vec[i]=0;
}
while (mes!=0){
    printf("ingresar la lluvia del mes %d: ",mes);
    scanf("%d",&lluvia);
    vec[mes]=vec[mes]+lluvia; //acumulo la cant de lluvia
    printf("ingresar el mes: ");
    scanf("%d",&mes);
}

i=0;
for(i=0;i<12;i++){
    //muestro lo acumulado mes por mes
    printf("Las lluvias del mes %d, fueron %d\n",i,vec[i]);
}
```

The screenshot shows a terminal window with the following output:

```
D:\Dropbox\UA\grado\ProgramacionEstructurada\Compartido PE\Unidades Te4r
ingresar el mes: 1
ingresar la lluvia del mes 1: 100
ingresar el mes: 1
ingresar la lluvia del mes 1: 200
ingresar el mes: 2
ingresar la lluvia del mes 2: 150
ingresar el mes: 0
Las lluvias del mes 0, fueron 0
Las lluvias del mes 1, fueron 300
Las lluvias del mes 2, fueron 150
Las lluvias del mes 3, fueron 0
Las lluvias del mes 4, fueron 0
Las lluvias del mes 5, fueron 0
Las lluvias del mes 6, fueron 0
Las lluvias del mes 7, fueron 0
Las lluvias del mes 8, fueron 0
Las lluvias del mes 9, fueron 0
Las lluvias del mes 10, fueron 0
Las lluvias del mes 11, fueron 0

-----
Process exited after 7.792 seconds with return value 0
Presione una tecla para continuar . . .
```

Estructuras estáticas - Arrays

Carga indirecta

- Se utiliza un valor como índice para diferentes vectores según la necesidad

Ejemplo

Una empresa vende 100 productos, necesitamos saber el total de productos vendidos y el importe acumulado por cada uno.

```
int cantidadVendida[10];
int totalVendido[10];
int codProducto=0;
int importe=0;
int i=0;
for(i=0;i<10;i++){ //inicializo los arrays
    cantidadVendida[i]=0;
    totalVendido[i]=0;
}

printf("Ingrese código de producto: ");
scanf("%d",&codProducto);
while(codProducto!=0){
    printf("Ingrese importe de la venta: ");
    scanf("%d",&importe);

    cantidadVendida[codProducto]++; //sumo uno a la cantidad vendida
    totalVendido[codProducto]+=importe; //acumulo el total vendido

    printf("Ingrese código de producto: ");
    scanf("%d",&codProducto);
}

for(i=0;i<10;i++){ //inicializo los arrays
    printf("Para el producto %d se vendieron %d unidades por un total de %d\n",i,cantidadVendida[i],totalVendido[i]);
}
```

D:\Dropbox\UA\grado\ProgramacionEstructurada\Compartido PE\Unidades Teóricas\Unidad 5

```
Ingrese código de producto: 1
Ingrese importe de la venta: 100
Ingrese código de producto: 1
Ingrese importe de la venta: 40
Ingrese código de producto: 2
Ingrese importe de la venta: 500
Ingrese código de producto: 3
Ingrese importe de la venta: 400
Ingrese código de producto: 0
Para el producto 0 se vendieron 0 unidades por un total de 0
Para el producto 1 se vendieron 2 unidades por un total de 140
Para el producto 2 se vendieron 1 unidades por un total de 500
Para el producto 3 se vendieron 1 unidades por un total de 400
Para el producto 4 se vendieron 0 unidades por un total de 0
Para el producto 5 se vendieron 0 unidades por un total de 0
Para el producto 6 se vendieron 0 unidades por un total de 0
Para el producto 7 se vendieron 0 unidades por un total de 0
Para el producto 8 se vendieron 0 unidades por un total de 0
Para el producto 9 se vendieron 0 unidades por un total de 0
```


Estructuras estáticas - Arrays

Array de char

- **Las cadenas de caracteres** (string) son declaradas en C como **arrays de caracteres**.
- En C **no existe un tipo predefinido** para manipular cadenas de caracteres (string). Sin embargo, el estándar de C define algunas funciones de biblioteca para tratamiento de cadenas.
- Una **cadena en C es un array de caracteres de una dimensión** que termina con el carácter especial `'\0'` (cero).

```
char hola[]="hola";
char chau[4]={'c','h','a','u'};

for(int x=0; x<4; x++){
    printf("el dato es %c\n",hola[x]);
}

for(int x=0; x<4; x++){
    printf("el dato es %c\n",chau[x]);
}
```



```
D:\Dropbox\UA\g
el dato es h
el dato es o
el dato es l
el dato es a
el dato es c
el dato es h
el dato es a
el dato es u
```

Estructuras estáticas - Arrays

Strings

- Se puede definir un **array dinámico** por medio de un puntero para poder utilizar array de cadenas de caracteres

```
char* nombres[3]; //defino un array de 3 posiciones

nombres[0]="luis";
nombres[1]="ana";
nombres[2]="berta";
for(int x=0; x<3; x++){
    printf("el dato es %s\n",nombres[x]);
}
```



D:\Dropbox\UAI\grado\Pro

```
el dato es luis
el dato es ana
el dato es berta
```

Estructuras estáticas - Arrays

Ejemplo

- Una empresa procesa N facturas con los datos
 - Nro. de factura
 - Nro. de Vendedor
 - Importe
- Si existe 5 vendedores, calcular
 1. Caja del día
 2. Comisión acumulada por vendedor (la comisión es del 10% para todos los vendedores)
 3. Porcentual que representa lo vendido por cada caja (del total).
 4. Valor promedio de las facturas

Gracias!