* **PASO DE ARRAYS A FUNCIONES**

Un array se puede pasar a una función como argumento. Sin embargo, la manera en la que el array se pasa difiere mucho de la de una variable ordinaria. Para pasar un aray a una función, el nombre de array debe aparecer solo, sin corchetes ni índices, como un argumento real en la llamada de la función. El correspondiente argumento formal se escribe de la misma manera, pero debe ser declarado como un array en la declaración de los argumentos formales.

Cuando un array se pasa como argumento de una función, los valores de los elementos del array no son pasados a la función , sino que el nombre del array se interpreta como una dirección del primer elemento del array, la dirección de memoria conteniendo el primer elemento del array. Esta dirección se asigna al correspondiente argumento formal cuando se llama a la función. El argumento formal se convierte por tanto en un puntero al primer elemento del array.   
Los argumentos pasados de esta manera se dice que son pasados por referencia en vez de por valor.

* **Ejemplo de función con vector:**

/\* Prototipo de una función que tiene como argumento un array de tipo entero\*/

float media(int vec[]);

A continuación se muestra un programa sencillo que pasa un array de 3 elementos enteros a una función donde se modifican dichos elementos.

#include<stdio.h>

#include <conio.h>

void modificar(int a[]); //Declaración de la función

main()

{

int cont, a[3]; /\*Declaración de array\*/

printf("\n Desde main, antes de llamar al a función \n");

for (cont=0; cont<=2; cont++)

{

a[cont]=cont+1;

printf("a[%d]= %d \n", cont, a[cont]);

}

modificar(a); //Llamada a la función

printf("\nDesde el main, después de llamar a la función: \n");

for(cont=0; cont<=2; cont++)

printf("a[%d] =%d\n", cont, a[cont]);

getchar(); }

void modificar(int a[]) /\*Definición de la función\*/

{

int cont;

printf("\n Desde la función, después de modificar los valores\n");

for(cont=0; cont<=2; cont++)

{

a[cont]=-9;

printf ("a[%d] = %d\n",cont, a[cont]);

}

return;

}

*La salida de este programa, es la siguiente:*

Desde main, antes de llamar a la función: a[0]=1 a[1]=2 a[2]=3

Desde main, después de llamar los valores: a[0]=-9 a[1]=-9 a[2]=-9

Desde main, después de llamar a la función: a[0]=-9 a[1]=-9 a[2]=-9

**Nota:**

Como ya se mencionó, los argumentos se pasan a una función por valor cuando los argumentos son variables ordinarias. Sin embargo, cuando se pasa un array a una función, no se pasan a la función los valores de los elementos del array. En su lugar, el nombre del array se interpreta como la dirección del primer elemento del array (la dirección de la posición de memoria que contiene el primer elemento del array). Esta dirección se asigna al correspondiente argumento formal cuando se llama a la función. El argumento formal se convierte por tanto en un puntero al primer elemento del array. Los argumentos pasados de esta manera se dice que son pasados por referencia en vez de por valor.

**Ejercicios:**

1. Elaborar un programa que tenga dos funciones: una que cree un vector de 5 elementos y otra que obtenga el número mayor y el número menor que hay en ese vector.
2. Por parejas, realicen 2 programas uno que sea de paso de parámetros por referencia con una variable normal y otro que sea de paso de parámetros por referencia con un vector.

* **Matríces como parámetros de funciones.**

Cuando pasamos un array como parámetro, en realidad estamos pasando un puntero al primer elemento del array, así que las modificaciones que hagamos en los elementos del array dentro de la función serán permanentes aún después de retornar.

Al pasar una matríz a una función, no se envía el número de filas, porque está tomando la dirección del primer elemento del array, pero si se debe enviar el número de columnas que tiene la matriz para que sepa en donde terminará.

* **Ejemplo de Función con Matríz:**

/\*Programa que calcula la suma de los elementos de dos tablas de números enteros

utilizando funciones (Funciones con Matrices)

Autor: Rosy Avendaño López

Fecha: Miércoles 5 de Febrero del 2014

\*/

#include<stdio.h>

#include<conio.h>

#define MAXFIL 20

#define MAXCOL 30

/\*declaración o prototipo de funciones\*/

void leerentrada (int t1[][MAXCOL], int, int);

void calcularsuma (int t1[][MAXCOL], int t2[][MAXCOL], int tr[][MAXCOL], int nfilas, int ncols);

void escribirsalida (int t1[][MAXCOL], int, int ncols);

/\*Programa principal\*/

main()

{

int nfilas, ncols;

/\*declaracion de los arrays\*/

int t1[MAXFIL][MAXCOL],t2[MAXFIL][MAXCOL],tr[MAXFIL][MAXCOL];

printf("Cuantas filas? ");

scanf("%d", &nfilas);

printf("Cuantas columnas? ");

scanf("%d", &ncols);

/\*Introducir datos para la primera tabla\*/

printf("\n Tabla 1 \n");

leerentrada(t1,nfilas,ncols); /\*Va a la Función para leer los datos de t1\*/

/\*Introducir datos para la segunda tabla\*/

printf("\n Tabla 2 \n");

leerentrada(t2,nfilas,ncols); /\*Va a la Función para leer los datos de t2\*/

calcularsuma(t1,t2,tr,nfilas,ncols); //Va a la Función para sumar las tablas

printf("\n \nTabla 1\n");

escribirsalida(t1, nfilas, ncols); //Va a la Función que imprime la tabla enviada (t1)

printf("\n \nTabla 2\n");

escribirsalida(t2, nfilas, ncols); //Va a la Función que imprime la tabla enviada (t2)

printf("\n \nLos resultados son:\n");

escribirsalida(tr, nfilas, ncols); //Va a la Función que imprime la tabla resultante (tr)

getch();

}

/\*Función que Lee una tabla de enteros\*/

void leerentrada(int t1[][MAXCOL], int m, int n)

{

int fila=0, col=0;

/\*Introducir datos para la tabla\*/

for (fila=0; fila<m; fila++)

for (col=0; col<n; col++)

{

printf("\nIntroducir datos para la fila %2d columna %2d: ",fila+1,col+1); scanf("%d",&t1[fila][col]);

}

}

/\*Funcion que Suma dos tablas de enteros\*/

void calcularsuma (int t1[][MAXCOL], int t2[][MAXCOL], int tr[][MAXCOL], int m, int n)

{

int fila, col;

/\*Realiza la suma y la guarda en la tabla de resultados\*/

for (fila=0; fila<m; fila++)

for (col=0; col<n; col++)

tr[fila][col]=t1[fila][col]+t2[fila][col];

}

/\*Función que imprime en pantalla una tabla de enteros\*/

void escribirsalida(int tr[][MAXCOL], int m, int n)

{

int fila, col;

/\*Imprimir los valores de la tabla\*/

for (fila=0; fila<m; fila++)

{

for (col=0; col<n; col++)

printf("%4d ", tr[fila][col]);

printf("\n");

}

}

Al ejecutar el programa se visualizará de la siguiente manera:

