	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:	Adrian Ulises Mercado Martínez
Asignatura:	Fundamentos de programación
Grupo:	07
No de Práctica(s):	09
Integrante(s):	Arredondo Granados Gerardo Casillas Herrera Leonardo Didier Diaz Gonzalez Rivas Angel Iñaqui Galván Castro Martha Selene
No. de Equipo de cómputo empleado:	20
No. de Lista o Brigada:	04
Semestre:	2022-1
Fecha de entrega:	25 de diciembre del 2021
Observaciones:	

CALIFICACIÓN:_____

Práctica 9
Arreglos unidimensionales

Índice

Objetivo

El alumno utilizará arreglos de una dimensión en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, alineados en un vector o lista.

Introducción

En la programación, se les llama arreglos a un conjunto de datos contiguos del mismo tipo, a estos arreglos se les asigna un tamaño definido al momento que nosotros los creamos. A los datos dentro del arreglo se les llama elementos del arreglo, a los elementos del arreglo se les da una posición particular.

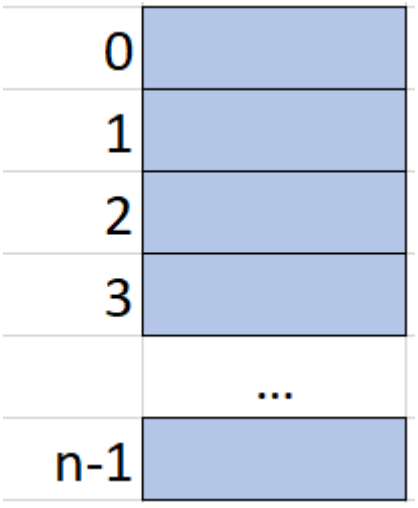
En lenguaje C tenemos arreglos unidimensionales y multidimensionales, la dimensión del arreglo va de acuerdo con el número de índices que se requiere para emplear para acceder a un elemento del arreglo. Por ejemplo, si nosotros queremos acceder a un elemento de un arreglo unidimensional, bastará con un índice, en cambio si queremos acceder a un elemento de un arreglo bidimensional, será necesario el uso de dos índices.

Nosotros como programadores, usamos los arreglos para eficientar el programa que se está haciendo, de igual manera como se usan para manipular datos del mismo tipo.

En esta práctica 9, nos enfocaremos en el trabajo de arreglos unidimensionales.

Arreglos Unidimensionales

Para tener en claro cómo es que se almacenan los elementos de un arreglo en la memoria de la computadora, lo podemos ver de la siguiente manera:



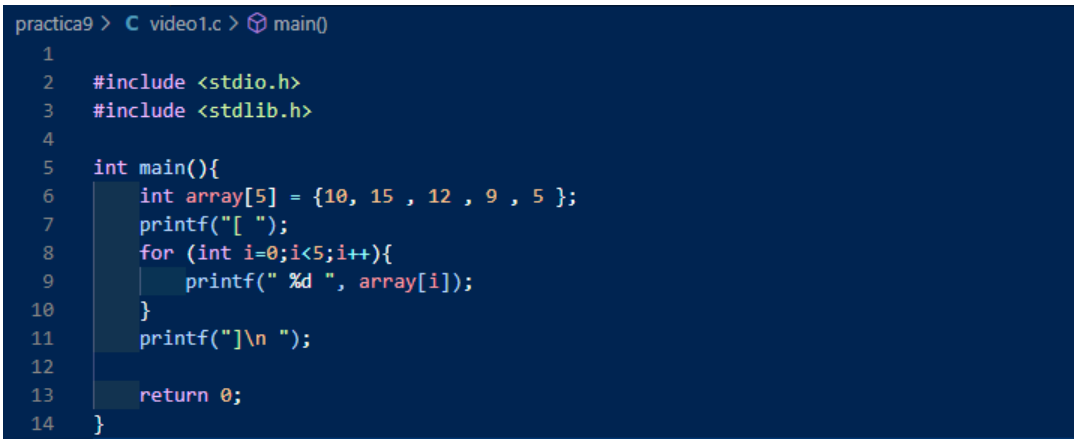
Teniendo en mente que **n** es el tamaño del arreglo, y este tamaño va de **0 a n-1**, se puede decir que la primera localidad en el arreglo le corresponde al índice 0, y la última le corresponde a n-1. Para definir un arreglo en lenguaje C, se hace de la siguiente manera:

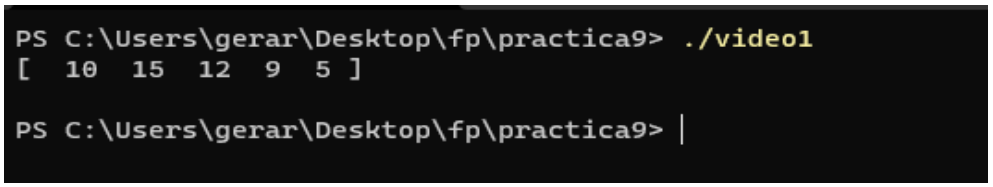
tipoDeDato nombre[tamaño]

Código

Video1.c

- Este programa genera un arreglo unidimensional de 5 elementos y posteriormente muestra los elementos del arreglo haciendo uso de un ciclo for





Video2.c

- Este programa le pide al usuario el tamaño del arreglo, después le pide los elementos del arreglo para después mostrar el arreglo. Se hace uso del ciclo for.

practica9 > C video2.c > ...

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

int main(){
    int tamA;
    printf("Cuantos elementos están en el arreglo?\n");
    scanf("%d", &tamA);

    int array[tamA];
    for (int i = 0;i <tamA;i++){
        printf("ingresa el valor para el elemento %d\n", i);
        scanf("%d", &array[i]);
    }
    printf("[ ");
    for (int i=0;i<tamA;i++){
        printf(" %d ", array[i]);
    }
    printf("]\n ");


    return 0;
}
```

```
PS C:\Users\gerar\Desktop\fp\practica9> ./video2
Cuantos elementos estan en el arreglo?
3
ingresa el valor para el elemento 0
1
ingresa el valor para el elemento 1
2
ingresa el valor para el elemento 2
3
[  1  2  3 ]

PS C:\Users\gerar\Desktop\fp\practica9>
```

Video3.c

- Este programa consta de un arreglo unidimensional de 5 elementos, el programa nos da la dirección del arreglo y después nos da la dirección de cada uno de los 5 elementos. Hacemos uso de un ciclo for.

practica9 > C video3.c > main()

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

int main(){
    int array[5];

    printf("La direccion del arreglo es: %p\n", &array);

    for(int i=0; i<5; i++){
        printf("La direccion del elemento %d del arreglo: %p\n", i, &array[i]);

    }


    return 0;
}
```

```
PS C:\Users\gerar\Desktop\fp\practica9> ./video3
La direccion del arreglo es: 0061FF08
La direccion del elemento 0 del arreglo: 0061FF08
La direccion del elemento 1 del arreglo: 0061FF0C
La direccion del elemento 2 del arreglo: 0061FF10
La direccion del elemento 3 del arreglo: 0061FF14
La direccion del elemento 4 del arreglo: 0061FF18
PS C:\Users\gerar\Desktop\fp\practica9>
```

Video5.c

- En este programa tenemos un arreglo unidimensional de 5 elementos, el programa imprimirá los 5 elementos usando apuntadores de dos maneras distintas. Se hace uno del ciclo for.

practica9 > C video5.c > main()

```c
//un apuntador es una variable que almacena la direccion de memoria para otra variable
// se indican con un * antes del nombre de la variable

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

int main(){
    int array[5]={1, 2,3,4,5};
    int *ptr_array = &array;

    for(int i=0; i<5;i++){
        printf("%d ", ptr_array[i]);
    }
    printf("\n");

// segunda manera de imprimir
    for(int i=0;i<5;i++){
        printf("%d ", *(ptr_array+i));
    }

    return 0;
}
```

```
PS C:\Users\gerar\Desktop\fp\practica9> ./video5
1 2 3 4 5
1 2 3 4 5
PS C:\Users\gerar\Desktop\fp\practica9>
```

Video5_2.c

- Al igual que en el programa anterior, tendremos un arreglo de 5 elementos, ahora en vez de imprimir los elementos, el programa nos dará las direcciones de cada uno de los elementos. Este programa igualmente usa apuntadores y ciclos for.

practica9 > C video5_2.c > ...

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

int main(){
    int array[5]={1, 2,3,4,5};
    int *ptr_array = &array;

    for(int i=0; i<5;i++){
        printf("%p\n ", &ptr_array[i]);
    }
    printf("\n");

// segunda manera de imprimir
    for(int i=0;i<5;i++){
        printf("%d ", (ptr_array+i));
    }

    return 0;
}
```

```
PS C:\Users\gerar\Desktop\fp\practica9> ./video5_2
0061FF00
 0061FF04
 0061FF08
 0061FF0C
 0061FF10

6422272 6422276 6422280 6422284 6422288
PS C:\Users\gerar\Desktop\fp\practica9>
```

Video7.c

- En este programa se le pedirá al usuario que ingrese el tamaño del arreglo, después se le pedirá el valor de los elementos del arreglo para después imprimir la dirección de estos elementos, todo esto usando apuntadores.

practica9 > C video7.c > main()

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

int main(){
    int tamA;
    int *ptr_array;
    printf("cuantos elementos tiene el arreglo?\n");
    scanf("%d", &tamA);

    ptr_array = (int *)malloc(tamA * sizeof(int));

    for(int i=0;i<tamA;i++){
        printf("ingresa el valor para elemento %d\n", i);
        scanf("%d", &ptr_array[i]);
    }
    printf("[ ");
    for(int i=0;i<tamA;i++){
        printf("%d ", ptr_array);
    }
    printf("]\n");
    free(ptr_array);
    ptr_array=NULL;
    return 0;
}
```

```
PS C:\Users\gerar\Desktop\fp\practica9> ./video7
cuantos elementos tiene el arreglo?
5
ingresa el valor para elemento 0
1
ingresa el valor para elemento 1
2
ingresa el valor para elemento 2
3
ingresa el valor para elemento 3
4
ingresa el valor para elemento 4
5
[ 8721328 8721328 8721328 8721328 8721328 ]
PS C:\Users\gerar\Desktop\fp\practica9>
```

Video8.c

- Este programa nos pedirá el tamaño del arreglo, se le pedirá al usuario ingresar el valor de los elementos para después imprimirlos en la pantalla. Cabe aclarar que este programa está elaborado con aritmética de apuntadores.

practica9 > C video8.c > main()

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

int main(){
    int tamA;
    int *ptr_array;
    printf("cuantos elementos tiene el arreglo?\n");
    scanf("%d", &tamA);

    ptr_array = (int *)malloc(tamA * sizeof(int));

    for(int i=0;i<tamA;i++){
        printf("ingresa el valor para elemento %d\n", i);
        scanf("%d", &ptr_array[i]);
    }
    printf("[ ");
    for(int i=0;i<tamA;i++){
        printf("%d ", *(ptr_array+i));
    }
    printf("]\n");
    free(ptr_array);
    ptr_array=NULL;
    return 0;
}

```

```
PS C:\Users\gerar\Desktop\fp\practica9> ./video8
cuantos elementos tiene el arreglo?
3
ingresa el valor para elemento 0
5
ingresa el valor para elemento 1
3
ingresa el valor para elemento 2
1
[ 5 3 1 ]
PS C:\Users\gerar\Desktop\fp\practica9>
```

Desarrollo

Actividades:

- Elaborar programas en lenguaje C que empleen arreglos de una dimensión
- Manipular este tipo de arreglos a través de índices

Ejercicio 1

- Crear la función para leer un arreglo utilizando el siguiente código

```
/**
 * () son corchetes no  parentesis
 */
void leer_arreglo(int a(), int n){
    for(int i= 0; i < n; i++){
        printf("Ingrese elemento %d\n", i);
        scanf("%d", &a(i));
    }
}
```

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


void leer_arreglo(int a[], int n);

int main(){
    leer_arreglo(5, 10);
    return 0;
}

void leer_arreglo(int a[], int n){
    for(int i = 0; i < n; i++){
        printf("Ingrese el elemento %d\n", i+1);
        scanf("%d", &a[i]);
    }
}
```

Ejercicio 2

- Crear la función para leer un arreglo utilizando el siguiente código

```
/**
 * () son corchetes no  parentesis
 */
void imprimir_arreglo(int a(), int n){
    for(int i= 0; i < n; i++){
        printf("Elemento (%d)=%d\n", i+1, a(i));
    }
}
```

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

void imprimir_arreglo(int a[], int n);

int main(){
    imprimir_arreglo(5, 10);
    return 0;
}

void imprimir_arreglo(int a[], int n){
    for(int i = 0; i < n; i++){
        printf("Elemento (%d) = %d\n", i+1, a[i]);
    }
}
```

Ejercicio 3

- Elaborar una función que invierta un arreglo

```c
void array_read(int a[], int n)
{
    for(int i = 0; i < n; i++){
        printf("Ingresa el valor del elemento %d:\n", i + 1);
        scanf("%d", &a[i]);
    }
```

```
    printf("Se terminó de leer el arreglo\n" );
}

void array_print(int a[], int n)
{
    printf("[");
    for(int i = 0; i < n; i++){
        printf("%d", a [i]);
    }
    printf("]\n");
}

void array_invert(int a[], int n){
    int temp;
    for(int i=0; j= n-1; i<(n/2); i++, j--){
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
    }
    array_print(a, n);
}
```

Ejercicio 4

- Elaborar una función que encuentre el número máximo de un arreglo

```c
void array_read(int a[], int n)
{
    for(int i = 0; i < n; i++){
        printf("Ingresa el valor del elemento %d:\n", i + 1);
        scanf("%d", &a[i]);
    }
    printf("Se terminó de leer el arreglo\n" );
}

void array_print(int a[], int n)
{
    printf("[");
    for(int i = 0; i < n; i++){
        printf("%d", a [i]);
    }
    printf("]\n");
}
int array_max(int a[], int n){
    int max = a[0];
    for(int i = 1; i <n ; i++){
        if(max < a[i]){
            max = a[i];
        }
    }
    printf("%d\n", max);
    return max;
}
```

Ejercicio 5

- Elaborar las mismas funciones pero con la versión de uso de memoria dinámica.

```c
void array_print_ptr(int *a, int n)
{
    for(int i = 0; i < n; i++){
        printf("Ingresa el valor del elemento %d:\n", i + 1);
        scanf("%d", (a+i));
    }
    printf("Se terminó de leer el arreglo\n" );
}

void array_print_ptr(int *a, int n)
{
    printf("[");
    for(int i = 0; i < n; i++){
        printf("%d", *(a+i));
    }
    printf("]\n");
}
void array_invert_ptr(int *a, int n){
    int temp;
    for(int i=0, j= n-1; i<(n/2); i++, j--){
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
    }
    array_print(a, n);
}
int array_max_ptr(int *a, int n){
    int max = a[0];
    for(int i = 1; i <n ; i++){
        if(max < a[i]){
            max = a[i];
        }
    }
    printf("%d\n", max);
    return max;
}
```

Conclusiones

Conclusión Grupal

Al desarrollar la práctica, se utilizaron los arreglos unidimensionales para mostrar una lista o un vector de valores del mismo tipo, con estos valores se resolvieron problemas utilizando funciones. De la misma manera, desarrollamos programas con el uso de la memoria dinámica a partir de apuntadores.

Conclusión individual

Arredondo Granados Gerardo: Los objetivos se cumplieron correctamente, en este práctica se realizaron programas con arreglos de una dimensión, dentro de los programas, aparte de hacer uso de los arreglos como lo marca la práctica, se tuvo que hacer uso de igual manera de los conocimientos adquiridos en las prácticas anteriores, todo esto para tener el programa mucho más completo y de una mejor manera. Para finalizar, el profesor nos indicó algunos problemas que debíamos resolver igualmente con arreglos, se resolvieron completamente dando por concluida la práctica 9.

Casillas Herrera Leonardo Didier: Aprendí a utilizar los arreglos unidimensionales para realizar distintas funciones, resolvimos los problemas planteados y agrupamos los datos del mismo tipo en el arreglo unidimensional utilizándolos como un vector; también lo hicimos con la memoria dinámica para buscar otra manera de usar los arreglos por lo que el objetivo de la práctica se cumplió.

Diaz Gonzalez Rivas Angel Iñaqui: En la práctica utilizamos arreglos para mostrar una lista de valores del mismo tipo en un programa. De la misma manera se resolvieron problemas que requerían agrupar datos del con el tipo de variable que le correspondía en un arreglo unidimensional. También se utilizó la aritmética de apuntadores para comprender su funcionamiento y poder utilizarlos de la forma adecuada.

Galván Castro Martha Selene: En conclusión en esta práctica se puede decir que cumplió con el objetivo de la práctica ya quedó claro el uso de arreglos unidimensionales ya que por medio de programas aprendimos su uso en la resolución de problemas y la manipulación de códigos para emplear el uso de memoria dinámica en los arreglos.

Referencias

- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.