



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: TISTA GARCÍA EDGAR

Asignatura: ESTRUCTURA DE DATOS Y ALGORITMOS I

Grupo: 1

No de Práctica(s): PRÁCTICA #2 APLICACIONES DE APUNTADORES

Integrante(s): CARRILLO CERVANTES IVETTE ALEJANDRA

*No. de Equipo de
cómputo empleado:* TRABAJO EN CASA

No. de Lista o Brigada: 9

Semestre: 2021 - 2

Fecha de entrega: 19 MARZO 2021

Observaciones:

CALIFICACIÓN: _____

PRÁCTICA #2 APLICACIONES DE APUNTADORES

Objetivo: Utilizar apuntadores en lenguaje C para acceder a las localidades de memoria tanto de datos primitivos como de arreglos.

Objetivo de la clase: Utilizar apuntadores para modificar valores de variables indirectamente, así como para realizar el manejo de arreglos y el paso por referencia en funciones

Ejemplos de la guía

Apuntador a carácter

La función principal de este programa es obtener, mediante un apuntador, la dirección de memoria de un carácter, así como su código ASCII. Estos datos son los que se verán como salida, aparte del carácter declarado, en este caso "x".

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>gcc ejemplo1.c -o ejemplo1.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>ejemplo1.exe

Carácter: x
Código ASCII: 120
Dirección de memoria: 6422299

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>_
```

ejemplo1.c

Apuntador a entero

Este programa, a partir de un apuntador de tipo entero, modifica la información de las variables declaradas "a" y "b", dando como salida los nuevos valores de estas y una pequeña explicación de qué operación se hizo en cada una.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>gcc ejemplo2.c -o ejemplo2.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>ejemplo2.exe

a = 5
b = 5 /*apEnt
b = 6 /*apEnt+1
a = 0 /*apEnt = 0

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>
```

ejemplo2.c

Apuntador a arreglo

En este programa se crea un apuntador de tipo entero que apunta al inicio de un arreglo, tiene como función imprimir la dirección en memoria del arreglo en la posición [0], del arreglo y del apuntador, lo cual se muestra en pantalla.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>gcc ejemplo3.c -o ejemplo3.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>ejemplo3.exe
Dirección del arreglo en la primera posición: 61ff12
Dirección del arreglo: 61ff12
Dirección del apuntador: 61ff12

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>
```

ejemplo3.c

Paso de variables por valor y por referencia

En este programa se inicializan los valores de un arreglo de 5 elementos, se declara un apuntador, el cual se encargará de modificar los datos de dicho arreglo; cuenta con 2 funciones, una la cual hará que el programa tenga paso por valor (donde al regresar a la función principal, no guarde ningún valor) y la otra hará que el programa tenga paso por referencia (donde al regresar a la función principal, se guarden los datos obtenidos).

Dado que el apuntador modificará un arreglo, las funciones utilizadas (paso por valor y paso por referencia) no deben de devolver valores, pero si deben de recibir parámetros.

Como salida de este programa se mostrará en pantalla los valores obtenidos en paso por valor y paso por referencia, además se mostrará el valor final del apuntador.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>gcc ejemplo4.c -o ejemplo4.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>ejemplo4.exe
Pasar valor: 55
55
128
Pasar referencia: 55
55
128
Valor final: 128

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>
```

ejemplo4.c

Aritmética de direcciones

En este programa se declara un apuntador el cual se asocia con un arreglo, la función principal de este programa es imprimir la dirección del apuntador, y como este está asociado al arreglo se imprime el elemento [0] de este; después, lo que se va a imprimir será el valor de la suma de una localidad al inicio del arreglo, esto quiere decir, que si al apuntador le sumas 1, te dará el elemento [1] del arreglo, y así sucesivamente. Obtenemos la siguiente salida:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>gcc ejemplo5.c -o ejemplo5.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>ejemplo5.exe
*apArr = 91
*(apArr+1) = 28
*(apArr+2) = 73

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1><
```

ejemplo5.c

Arreglo unidimensional

La función de este programa es imprimir la dirección de memoria de los números que tiene un arreglo de 5 elementos, esto se hace con ayuda de un apuntador asociado a dicho arreglo y para imprimirse se utiliza un ciclo for con un printf %x, donde x es el apuntador más el contador del ciclo de repetición.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>gcc ejemplo6.c -o ejemplo6.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>ejemplo6.exe
61ff0e
61ff10
61ff12
61ff14
61ff16

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>
```

ejemplo6.c

Arreglo bidimensional

La función de este programa es parecida al código anterior, pero en este caso es con un arreglo bidimensional, lo que hace el programa es, mediante un apuntador (asociado al arreglo), imprime la dirección de memoria de los números inicializados en el dicho arreglo de [3][3] y como salida, te muestra:

```

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>gcc ejemplo7.c -o ejemplo7.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>ejemplo7.exe

61fef4 61fef8 61fefc
61ff00 61ff04 61ff08
61ff0c 61ff10 61ff14
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>

```

ejemplo7.c

Cifrado César

La función principal de este programa es Cifrar y Descifrar mensajes en Cifrado César, esto se realizó por medio del uso de 2 arreglos (uno donde se almacenan las variables del abecedario normal y otro donde se almacenan las variables del abecedario recorrido) y apuntadores, mediante un algoritmo que recorre la posición de las letras y así poder hacer el cifrado correctamente, para realizar este programa se utilizaron 2 funciones aparte de la principal, una función para cifrar y otra para descifrar. Como salida queda:

```

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>gcc ejemplo8.c -o ejemplo8.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>ejemplo8.exe

*** CIFRADO CÉSAR ***
ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFGHIJKLMNOPQRSTUVWXYZABC

Elegir una opción:
1) Cifrar
2) Descifrar.
3) Salir.
1
Ingresar la palabra a cifrar (en mayúsculas): ALEJANDRA
El texto ALEJANDRA cifrado es: DOHMDQGUD

*** CIFRADO CÉSAR ***
ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFGHIJKLMNOPQRSTUVWXYZABC

Elegir una opción:
1) Cifrar
2) Descifrar.
3) Salir.
2
Ingresar la palabra a descifrar (en mayúsculas): DOHMDQGUD
El texto DOHMDQGUD descifrado es: ALEJANDRA

*** CIFRADO CÉSAR ***
ABCDEFGHIJKLMNOPQRSTUVWXYZ
DEFGHIJKLMNOPQRSTUVWXYZABC

Elegir una opción:
1) Cifrar
2) Descifrar.
3) Salir.
3

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>

```

ejemplo8.c

EJERCICIOS PROPUESTOS PARA EL DESARROLLO DE LA PRÁCTICA

Ejercicio 1

- Codifica el siguiente programa
- Explica y corrige los errores que encuentras
- Indica que hace y cuál es la salida en pantalla

El código tenía muchos errores al momento de utilizar los apuntadores, pues bien, como vimos en clase, para utilizar un apuntador debemos de seguir 3 pasos ej:

1. Declarar el apuntador

```
int *hola;    // Se usa el operador indirección
int var = 3;  // Otra variable
```

2. Asociación del apuntador con otra variable

```
hola = &var;  //Aquí ya no se utiliza el operador indirección, en cambio se utiliza el operador
              dirección
```

3. Utilizar el apuntador

```
*hola=5;  // Aquí se utiliza el operador indirección
```

al declarar los apuntadores, a uno le faltaba el operador indirección, después al asociar el apuntador con otra variable, les faltaba o sobraba el operador dirección, finalmente al usar el apuntador en alguna operación también tenían algunos errores en cuestión de los operadores, así que se modificaron:

```
int *p1, *p2, *p3, *p4; //apuntadores
//p3 es un apuntador, por lo tanto se debe de utilizar el operador indirección *
p1 = &w;
p2 = &x; //al asociar el valor con otra variable, el apuntador no debe de llevar el operador indirección
p3 = &y; //al asociar el valor con otra variable, el apuntador no debe de llevar el operador indirección
        // a la variable asociada se le debe de poner el operador dirección &
p4 = p1; //p1 no lleva el operador indirección
r1 = *p1 + *p2; //p1 debe de llevar el operador indirección *
r2 = *p3 * *p4; //p3 y p4 debe de llevar el operador indirección *
r3 = *p2 + *p3; //r3 no debe de llevar el operador indirección, ya que no es un apuntador
```

modificación

La función principal de este programa es hacer diferentes operaciones, con el uso de los apuntadores, y como salida, se imprimen los resultados de dichas operaciones con las variables modificadas.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>gcc ejercicio1.c -o ejercicio1.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>ejercicio1.exe

Los Resultados son 30, 300 y 50

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>
```

ejercicio1.c

Ejercicio 2

Codifica y compila el siguiente programa.

- Completa el programa para utilizar correctamente el arreglo de 4 dimensiones
- Indica cuáles son los índices del arreglo que se almacenan en las variables a, b y c. Comprueba tu razonamiento con alguna captura de pantalla
- Utilizando el apuntador y un solo ciclo, recorre el arreglo de tal manera que se modifiquen todos sus valores de la siguiente manera: plano 0, valores múltiplos de 5 (5,10,15,20); plano 1: valores múltiplos de 6 (6,12,18,...), plano 2, valores múltiplos de 7 (7,14,21,28...) plano 3, valores múltiplos de 3 (3,6,9,12)

En el inciso a), lo que hice para que el arreglo se utilizará completamente con 4 dimensiones, fue agregar un for con la dimensión que faltaba, la cual era [3] el “super plano”, le asigne como variable h y la puse en el orden correspondiente.

```
15     for (int h = 0; h < 3; ++h)
16     {
17         for (int i = 0; i < 4; ++i)
18         {
19             for (int j = 0; j < 5; ++j)
20             {
21                 for (int k = 0; k < 2; ++k)
22                 {
23                     arr1 [h][i][j][k] = var;
24                     var += 2;
25                 }
26             }
27         }
28     }
29 }
```

Inciso a)

En el inciso b), se me complicó al momento de poner los índices donde se almacenan las variables a, b, c, ya que estaba intentando poner un apuntador asociado a el arreglo y de ahí poner un ciclo for con un 4 if, else-if donde cada uno llevara el índice correspondiente y una condición de que si era el arreglo igual a la variable correspondiente, saliera el número donde se encontraba dicha variable, esto con ayuda de *(point); sin embargo, creo esa idea estuvo mal planteada y no logré hacer lo que solicitaba el ejercicio.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>gcc ejercicio2.c -o ejercicio2.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>ejercicio2.exe

a=7    b=25    c=55

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>_
```

Inciso b)

Finalmente, respecto al inciso c), tampoco pude terminarlo al 100%, ya que con un ciclo for, no me quedaba, pero recordé como estaba intentando hacer el inciso anterior y en este caso, me quedo algo parecido a lo solicitado


```

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>gcc ejercicio2.c -o ejercicio2.exe
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>ejercicio2.exe

a=7 b=25 c=55

5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105
110 115 120 125 130 135 140 145 150 6 12 18 24 30 36 42 48 54 100 105

105 110 115 120 125 130 135 140 145 150 6 12 18 24 30 36 42 48 54 60 66
72 78 84 90 96 102 108 114 120 126 132 138 144 150 156 162 168 174 180 186
14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133 140 147

66 72 78 84 90 96 102 108 114 120 126 132 138 144 150 156 162 168 174 180 7
14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133 140 147
154 161 168 175 182 189 196 203 210 3 6 9 12 15 18 21 24 27 30 33

7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133 140 147
154 161 168 175 182 189 196 203 210 3 6 9 12 15 18 21 24 27 30 33
36 39 42 45 48 51 54 57 60 63 66 69 72 75 78 81 84 87 90 93
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P2 V1>

```

Inciso c)

Lo que utilice para esta salida, fue un ciclo for que contara de 1 en 1 hasta llegar al 121, ya que al multiplicar los índices nos queda 120, entonces ese for, lo dividi en 4 partes (4 condicionales if) que es el número de planos requeridos, en cada if puse una condición de que número a que número quería que tuvieran múltiplos de 5, 6, 7, 3 respectivamente y dentro de esas condicionales, utilice el apuntador para que se cambiara el valor de cada elemento del arreglo original.

Finalmente imprimí el arreglo con dos for, pero no quedo bien acomodado y al tratar de modificarlo, modifique algo y ya no lo pude imprimir, pero antes de que eso pasara, tome una captura de como me quedaba anteriormente

Ejercicio 3.

Escribe un programa en el cual, se soliciten al usuario 2 valores (almacenadas en 2 variables) y mediante una función que no devuelva ningún valor y que utilice paso por referencia, se calcule lo siguiente:*

- El resultado de elevar el primero a la potencia que indique el segundo*
- La división del valor obtenido en el inciso a y el primer valor ingresado*

Ejemplo: si el usuario ingresa 8 y 4

- $8^4 = 4096$;*
- $4096 / 8 = 512$*

El resultado de a y b se debe almacenar en las mismas variables donde se almacenaron los datos ingresados por el usuario

**Nota: la función solicitada no debe contener ninguna instrucción "printf" ni ninguna variable auxiliar*

Este programa pide al usuario ingresar dos números los cuales se almacenarán en dos variables distintas y con ellos se realizará una operación matemática; la función principal de este programa, aparte de realizar la operación, es utilizar apuntadores con el fin de modificar dichas variables a través de una función, esta función será la encargada de realizar las operaciones antes ya mencionadas sin ocupar alguna variable auxiliar, por ello se hará uso del apuntador. Aparte nos dice que la función no debe de contener la instrucción "printf", ni debe de devolver valores, por lo cual sabemos que, para utilizar el apuntador en dicha función, deberá ser una función que si reciba parámetros pero que no devuelva valores.

Finalmente imprimimos en la función principal el resultado:


```

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>gcc ejercicio3.c -o ejercicio3.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>ejercicio3.exe

Ingrese valor 1: 4
Ingrese valor 2: 5

a) Resultado de elevar el primero a la potencia que indique el segundo: 1024
b) División del valor obtenido en el inciso a y el primer valor ingresado: 256

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>_

```

Ejercicio3.c

Ejercicio 4

Escribe un programa que mediante una función modifique los valores de un arreglo bidimensional de 5 x 4 con valores iniciales dados por el usuario, duplicando (multiplicando x 2) los valores de los renglones pares (0,2,4) y multiplicando x 3 los renglones impares (1,3)

En este programa primero utilice 2 ciclos for para pedirle al usuario que ingresara dato por dato del arreglo bidimensional, para que después con ayuda de un apuntador (el cual asocié con el arreglo) y una función que si recibe parámetros, pero no devuelve valores (por ser un arreglo `arr[i][j]`), poder modificarlo para que los renglones pares se multipliquen por 2 y los impares por 3, esto se hizo por medio un if-else

If – Si los renglones es 0, 2, 4, se multiplicarán por 2

Else- Si no son esos renglones (en caso de ser renglón 1, 3) se multiplicarán por 3

```

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>ejercicio4.exe

Ingrese valor [1] [1]: 14
Ingrese valor [1] [2]: 25
Ingrese valor [1] [3]: 3
Ingrese valor [1] [4]: 9
Ingrese valor [2] [1]: 2
Ingrese valor [2] [2]: 14
Ingrese valor [2] [3]: 12
Ingrese valor [2] [4]: 3
Ingrese valor [3] [1]: 5
Ingrese valor [3] [2]: 85
Ingrese valor [3] [3]: 75
Ingrese valor [3] [4]: 3
Ingrese valor [4] [1]: 6
Ingrese valor [4] [2]: 8
Ingrese valor [4] [3]: 45
Ingrese valor [4] [4]: 4
Ingrese valor [5] [1]: 8
Ingrese valor [5] [2]: 54
Ingrese valor [5] [3]: 45
Ingrese valor [5] [4]: 1

Arreglo Original:

14    25    3    9
2     14   12   3
5     85   75   3
6     8    45   4
8     54   45   1

Arreglo modificado:
Los valores de los renglones pares se multiplican por 2 y
los valores de los renglones impares por 3

28    50    6    18
6     42   36    9
10    170   150   6
18    24   135  12
16    108   90    2

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P2 V1>

```

Conclusiones

Se cumplieron los objetivos de esta práctica, ya que se utilizaron apuntadores para modificar variables indirectamente, como fue el caso de los diferentes ejercicios propuestos, también vimos el manejo de arreglos con apuntadores y funciones; en el caso del ejercicio 2 se vio mejor el uso de apuntadores para acceder a las localidades de memoria de arreglos.

No se logró cubrir el 100% de los ejercicios solicitados, ya que en el ejercicio 2, al momento de usar apuntadores para recorrer el arreglo, se me presento una dificultad, ya que solo era con un ciclo for, intente hacer algo parecido a los ejercicios de la guía pero no me quedo; sin embargo, los ejercicios 1, 3, 4, si se lograron cumplir al 100%. Considero que estos ejercicios si contribuyeron a los conceptos vistos en clase, ya que esta práctica solicitaba ejercicios donde se ocupaban todos los temas vistos anteriormente, como son los arreglos y las funciones.

Personalmente, considero que este tema (apuntadores) es un poco complejo ya que te puedes confundir al momento de escribir los códigos, como fue en mi caso, pero siento que es un tema de gran importancia por que te facilita la escritura del código. Estos ejercicios estuvieron muy bien planteados, ya que a pesar de que no logré terminar el ejercicio 2, me puse a investigar más acerca de los apuntadores y me quedo una idea un poco más clara de su función.

Referencias

- Joyanes. Programación en C. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software. 2da Edición