

Laboratorios de computación salas A y B

Marco Antonio Martinez Quintana

Profesor:

Fundamentos de Programación

Asignatura:

3

Grupo:

11

No de Práctica(s):

Enrique Ichazo Bautista

Integrante(s):

*No. de Equipo de
cómputo empleado:*

No aplica

27

No. de Lista o Brigada:

2021-1

Semestre:

11/01/2021

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

ARREGLOS UNIDIMENSIONALES Y MULTIDIMENSIONALES

Objetivo:

Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

Actividades:

Elaborar un programa en lenguaje C que emplee arreglos de una dimensión.

Resolver un problema que requiera el uso de un arreglo de dos dimensiones, a través de un programa en lenguaje C.

Manipular arreglos a través de índices y apuntadores.

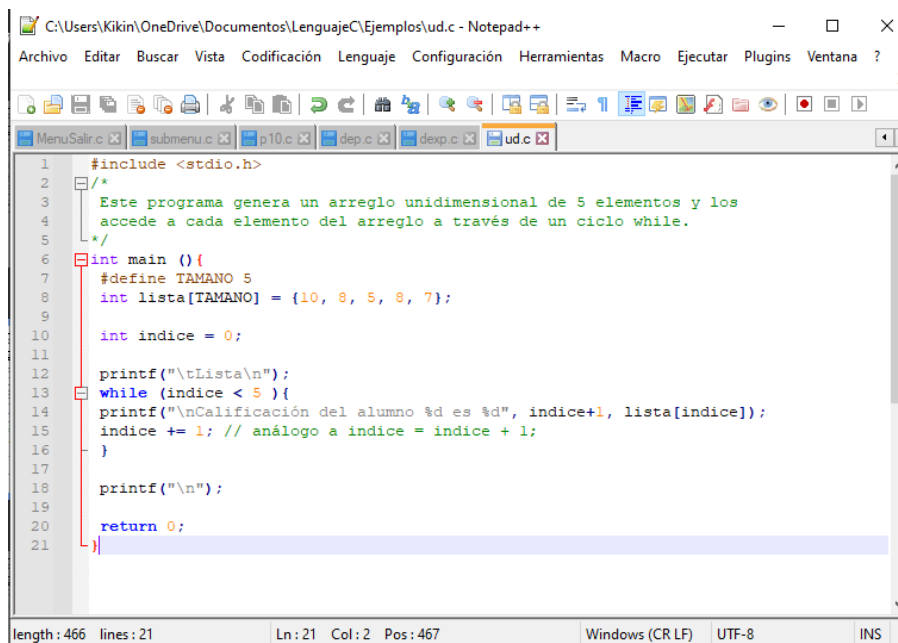
Introducción

Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse.

A cada elemento (dato) del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices.

Los arreglos pueden ser unidimensionales o multidimensionales. Los arreglos se utilizan para hacer más eficiente el código de un programa.

Arreglo unidimensional while



```
1 #include <stdio.h>
2
3 /*
4  Este programa genera un arreglo unidimensional de 5 elementos y los
5  accede a cada elemento del arreglo a través de un ciclo while.
6 */
7 int main () {
8     #define TAMANO 5
9     int lista[TAMANO] = {10, 8, 5, 8, 7};
10
11     int indice = 0;
12     printf("\tLista\n");
13     while (indice < 5) {
14         printf("\nCalificación del alumno %d es %d", indice+1, lista[indice]);
15         indice += 1; // análogo a indice = indice + 1;
16     }
17     printf("\n");
18     return 0;
19 }
```

length: 466 lines: 21 Ln: 21 Col: 2 Pos: 467 Windows (CR LF) UTF-8 INS

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.18363.1316]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Kikin>cd OneDrive
C:\Users\Kikin\OneDrive>cd Documentos
C:\Users\Kikin\OneDrive\Documentos>cd LenguajeC
C:\Users\Kikin\OneDrive\Documentos\LenguajeC>cd Ejemplos
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>gcc ud.c -o ud.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>ud.exe
Lista
Calificaci|n del alumno 1 es 10
Calificaci|n del alumno 2 es 8
Calificaci|n del alumno 3 es 5
Calificaci|n del alumno 4 es 8
Calificaci|n del alumno 5 es 7
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>_
```

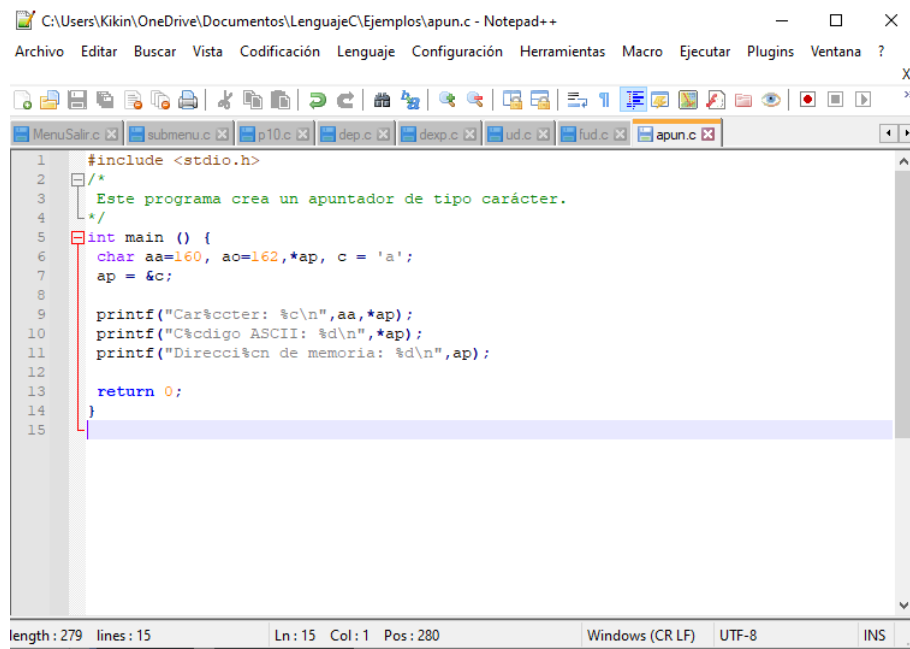
Arreglo unidimensional for

```
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos\fud.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?

1  #include <stdio.h>
2  /*
3   * Este programa genera un arreglo unidimensional de 5 elementos y
4   * accede a cada elemento del arreglo a través de un ciclo for.
5   */
6  int main () {
7      char
8      ao=162;
9      #define TAMANO 5
10     int lista[TAMANO] = {10, 8, 5, 8, 7};
11
12     printf("\tLista\n");
13     for (int indice = 0 ; indice < 5 ; indice++){
14         printf("\nCalificaci|cn del alumno %d es %d",ao, indice+1, lista[indice]);
15     }
16
17     printf("\n");
18
19     return 0;
20 }
```

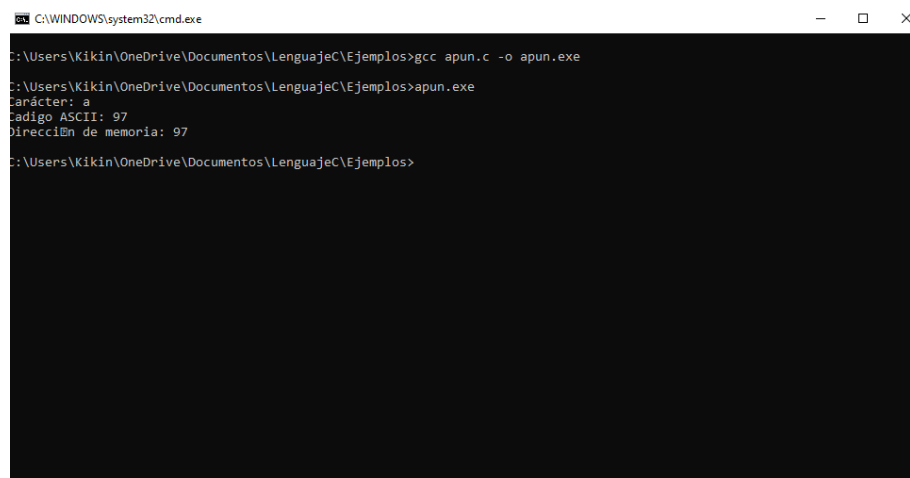
```
C:\WINDOWS\system32\cmd.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>gcc fud.c -o fud.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>fud.exe
Lista
Calificación del alumno 1 es 10
Calificación del alumno 2 es 8
Calificación del alumno 3 es 5
Calificación del alumno 4 es 8
Calificación del alumno 5 es 7
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>_
```

Código apuntores



```
1 #include <stdio.h>
2
3 /*
4  * Este programa crea un apuntador de tipo carácter.
5  */
6 int main () {
7     char aa=160, *ap=&aa, c = 'a';
8     ap = &c;
9
10    printf("Carácter: %c\n",aa,*ap);
11    printf("Código ASCII: %d\n",*ap);
12    printf("Dirección de memoria: %d\n",ap);
13
14    return 0;
15 }
```

length: 279 lines: 15 Ln: 15 Col: 1 Pos: 280 Windows (CR LF) UTF-8 INS



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>gcc apun.c -o apun.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>apun.exe
Carácter: a
Código ASCII: 97
Dirección de memoria: 97
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>
```

C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos\apt.c - Notepad++

Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?

```
4  //través de un apuntador.
5  */
6  int main () {
7      int a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};
8      int *apEnt;
9      apEnt = &a;
10
11     printf("a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}\n");
12     printf("apEnt = %a\n");
13
14     b = *apEnt;
15     printf("b = *apEnt \t-> b = %i\n", b);
16
17     b = *apEnt + 1;
18     printf("b = *apEnt + 1 \t-> b = %i\n", b);
19
20     *apEnt = 0;
21     printf("*apEnt = 0 \t-> a = %i\n", a);
22
23     apEnt = &c[0];
24     printf("apEnt = %c[0] \t-> apEnt = %i\n", *apEnt);
25
26     return 0;
27 }
```

length: 596 lines: 27 Ln: 26 Col: 11 Pos: 594 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

```
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>gcc apt.c -o apt.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>apt.exe
a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}
apEnt = &a
b = *apEnt      -> b = 5
b = *apEnt + 1  -> b = 6
*apEnt = 0      -> a = 0
apEnt = &c[0]   -> apEnt = 5
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>
```

C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos\apn2.c - Notepad++

Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?

```
1 #include <stdio.h>
2
3 /*
4  Este programa trabaja con aritmética de apuntadores para acceder a los
5  valores de un arreglo.
6 */
7 int main () {
8     int arr[] = {5, 4, 3, 2, 1};
9     int *apArr;
10    apArr = arr;
11
12    printf("int arr[] = {5, 4, 3, 2, 1};\n");
13    printf("apArr = %arr[0]\n");
14
15    int x = *apArr;
16    printf("x = *apArr \t -> x = %d\n", x);
17
18    x = *(apArr+1);
19    printf("x = *(apArr+1) \t -> x = %d\n", x);
20
21    x = *(apArr+2);
22    printf("x = *(apArr+1) \t -> x = %d\n", x);
23
24    return 0;
25 }
```

length: 489 lines: 25 Ln: 25 Col: 1 Pos: 490 Windows (CR LF) UTF-8 INS

C:\WINDOWS\system32\cmd.exe

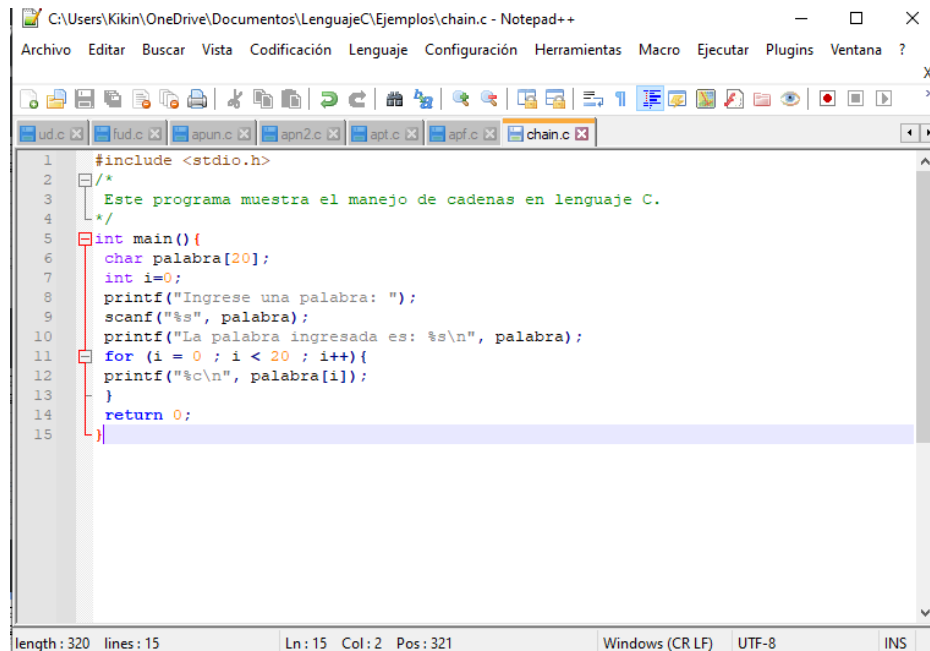
```
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>gcc apn2.c -o apn2.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>apn2.exe
int arr[] = {5, 4, 3, 2, 1};
apArr = &arr[0]
x = *apArr      -> x = 5
x = *(apArr+1)  -> x = 4
x = *(apArr+1)  -> x = 3
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>
```

Código apuntadores (for)

```
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos\apf.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
[ud.c] [fud.c] [apun.c] [apn2.c] [apt.c] [apf.c]
1  #include <stdio.h>
2  /*
3   * Este programa genera un arreglo unidimensional de 5 elementos y
4   * accede a cada elemento del arreglo a través de un apuntador
5   * utilizando un ciclo for.
6   */
7  int main () {
8      char
9      ao=162;
10     #define TAMANO 5
11     int lista[TAMANO] = {10, 8, 5, 8, 7};
12     int *ap = lista;
13     printf("\tLista\n");
14     for (int indice = 0 ; indice < 5 ; indice++){
15         printf("\nCalificaci%cn del alumno %d es %d",ao, indice+1, *(ap+indice));
16     }
17     printf("\n");
18
19     return 0;
20 }
21
length: 475  lines: 21  Ln: 21  Col: 2  Pos: 476  Windows (CR LF)  UTF-8  INS
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>gcc apf.c -o apf.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>apf.exe
Lista
Calificación del alumno 1 es 10
Calificación del alumno 2 es 8
Calificación del alumno 3 es 5
Calificación del alumno 4 es 8
Calificación del alumno 5 es 7
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>
```

Código apuntadores (cadenas)



```
1 #include <stdio.h>
2 /*
3  Este programa muestra el manejo de cadenas en lenguaje C.
4  */
5 int main(){
6     char palabra[20];
7     int i=0;
8     printf("Ingrese una palabra: ");
9     scanf("%s", palabra);
10    printf("La palabra ingresada es: %s\n", palabra);
11    for (i = 0 ; i < 20 ; i++){
12        printf("%c\n", palabra[i]);
13    }
14    return 0;
15 }
```

length: 320 lines: 15 Ln: 15 Col: 2 Pos: 321 Windows (CR LF) UTF-8 INS



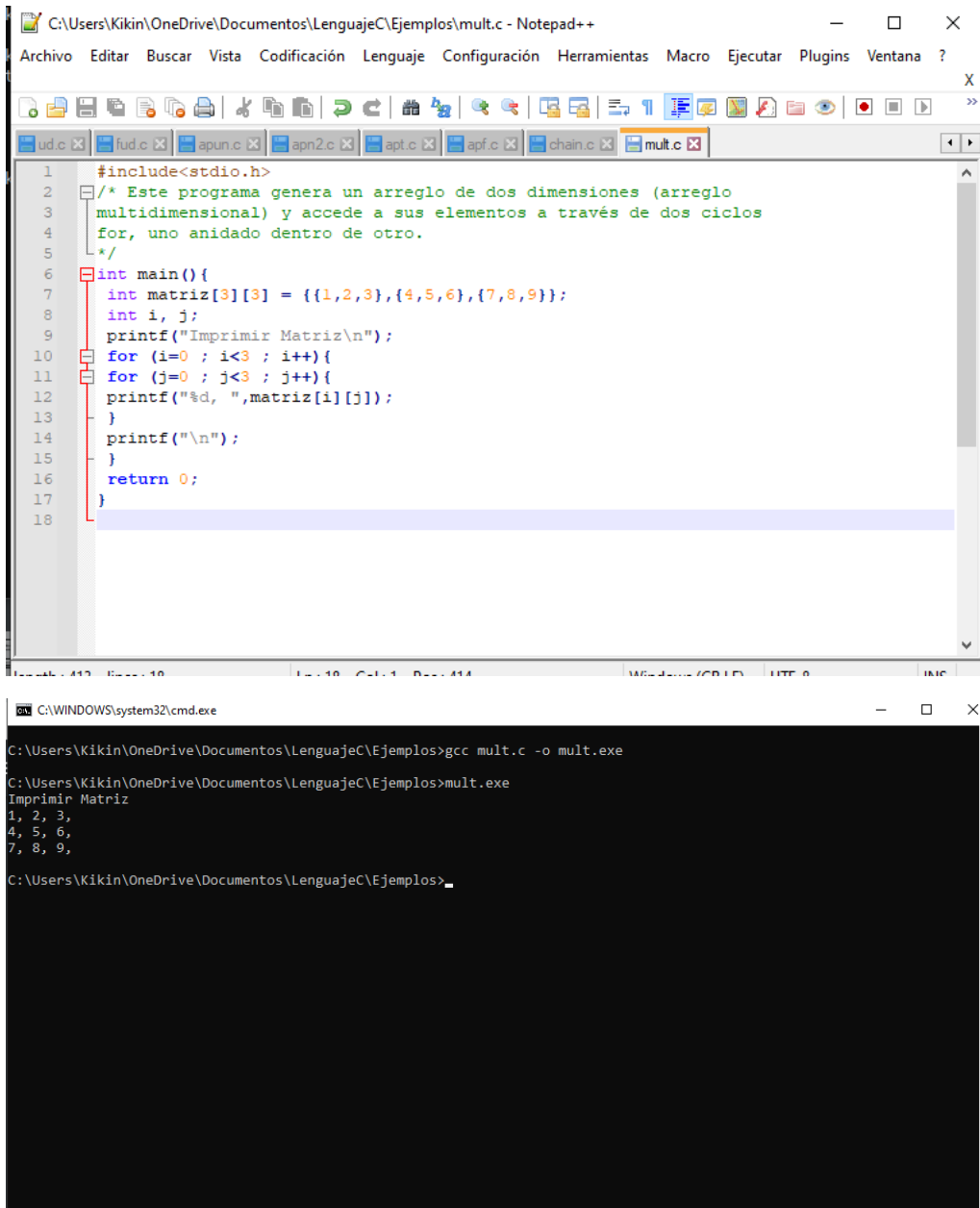
```
C:\WINDOWS\system32\cmd.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>gcc chain.c -o chain.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>chain.exe
Ingrese una palabra: bicho
La palabra ingresada es: bicho
b
i
c
h
o

@
P
@

á
!

C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>
```


Arreglos multidimensionales



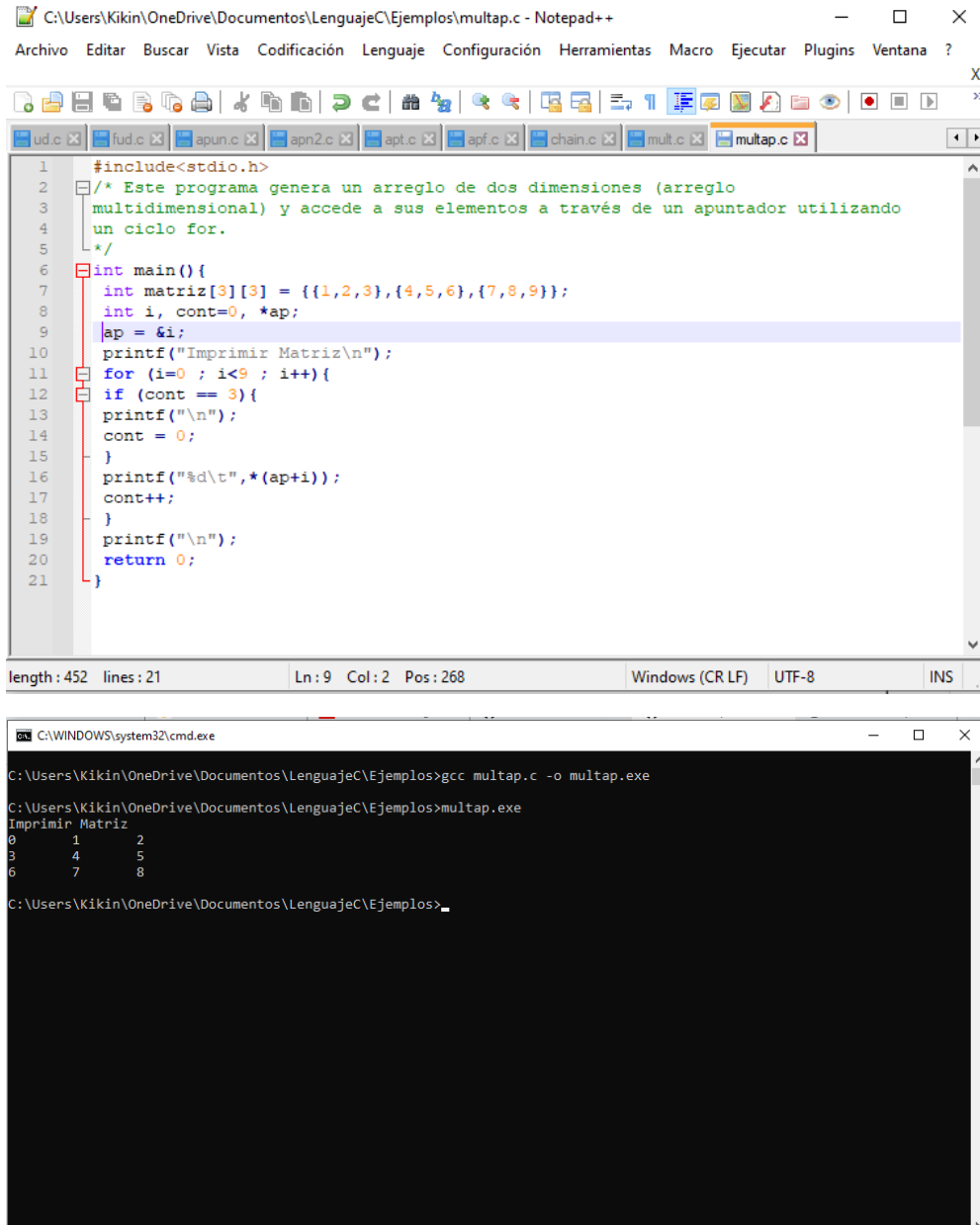
The image shows two windows. The top window is Notepad++ editing a file named 'mult.c'. The code defines a 3x3 matrix and prints its contents using nested loops. The bottom window is a Windows command prompt showing the compilation of 'mult.c' into 'mult.exe' using 'gcc', followed by running the executable, which outputs the matrix values.

```
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos\mult.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?

1  #include<stdio.h>
2  /* Este programa genera un arreglo de dos dimensiones (arreglo
3  multidimensional) y accede a sus elementos a través de dos ciclos
4  for, uno anidado dentro de otro.
5  */
6  int main(){
7      int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
8      int i, j;
9      printf("Imprimir Matriz\n");
10     for (i=0 ; i<3 ; i++){
11         for (j=0 ; j<3 ; j++){
12             printf("%d, ",matriz[i][j]);
13         }
14         printf("\n");
15     }
16     return 0;
17 }
18

C:\WINDOWS\system32\cmd.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>gcc mult.c -o mult.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>mult.exe
Imprimir Matriz
1, 2, 3,
4, 5, 6,
7, 8, 9,
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>
```

Arreglos multidimensionales con apuntadores



The image shows two windows. The top window is Notepad++ editing a file named 'multap.c'. The code is as follows:

```
1 #include<stdio.h>
2 /* Este programa genera un arreglo de dos dimensiones (arreglo
3 multidimensional) y accede a sus elementos a través de un apuntador utilizando
4 un ciclo for.
5 */
6 int main() {
7     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
8     int i, cont=0, *ap;
9     ap = &i;
10    printf("Imprimir Matriz\n");
11    for (i=0 ; i<9 ; i++){
12        if (cont == 3){
13            printf("\n");
14            cont = 0;
15        }
16        printf("%d\t",*(ap+i));
17        cont++;
18    }
19    printf("\n");
20    return 0;
21 }
```

The bottom window is a Windows command prompt running the program. The output is:

```
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>gcc multap.c -o multap.exe
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>multap.exe
Imprimir Matriz
0      1      2
3      4      5
6      7      8
C:\Users\Kikin\OneDrive\Documentos\LenguajeC\Ejemplos>
```

Conclusión:

Creo que estos arreglos nos servirán para ser más ordenados e identificar de manera sencilla nuestras condiciones, supongo que servirá para programar bases de datos, me gustaría añadir que tuve que depurar el último ejemplo porque había un error en el renglón 9, así que creo que el curso me ha llevado a convertirme en un

programador más completo conforme avanza, por lo tanto puedo asegurar que el curso me ha dado las bases de las cuales carecía al inicio del semestre y considero que esté cumplió sus objetivos.

Bibliografía

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