



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. MARCO ANTONIO MARTÍNEZ QUINTANA

Asignatura: FUNDAMENTOS EN COMPUTACIÓN

Grupo: 1103

No de Práctica(s): # 11

Integrante(s): RAUDALES PALMA LEONARDO DE JESÚS

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada:

Semestre: PRIMER SEMESTRE

Fecha de entrega: 15 DE ENERO 2021

Observaciones:

CALIFICACIÓN: _____

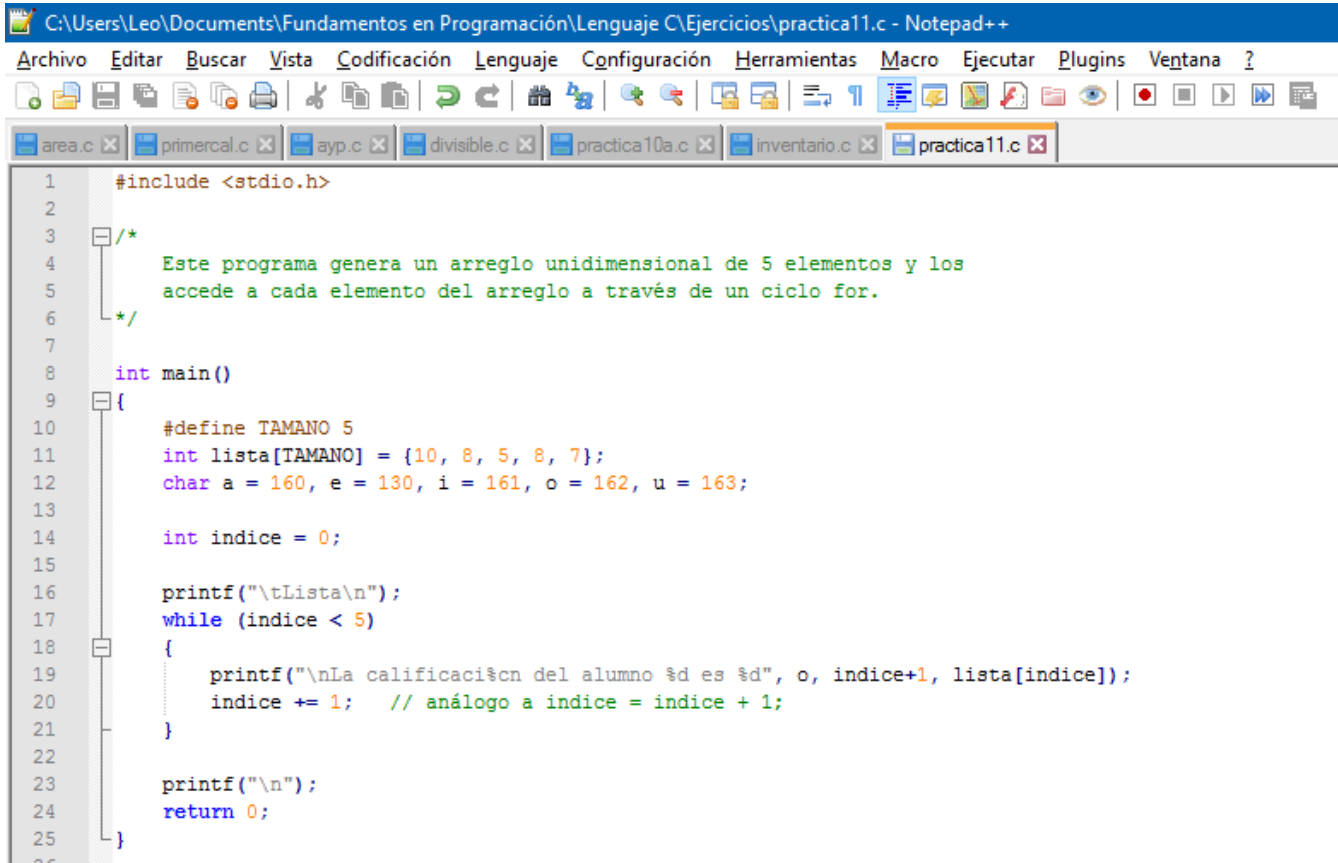
INTRODUCCIÓN:

Los arreglos son una herramienta importante en el rubro de la programación, sobretodo sirven para reservar espacios de memoria para diferente número de datos, así podemos agrupar datos en conjuntos de diferentes dimensiones, ya sean unidimensionales, es decir, arreglos de una sola columna o un solo renglón, o multidimensionales que son arreglos más complejos donde se puede almacenar más cantidad de datos, estos poseen múltiples columnas y renglones con espacios que van ocupando cada dato solicitado en la variable del arreglo.

Dentro de los arreglos también existe lo que es un apuntador, estos son llamados así porque muestran la dirección dentro de la memoria donde se almacena alguna variable con su valor, por supuesto, esta variable puede ser un arreglo, por lo cual, un apuntador también puede mostrar en donde es que se encuentran los datos de dicho arreglo en cuestión, sea unidimensional o multidimensional.

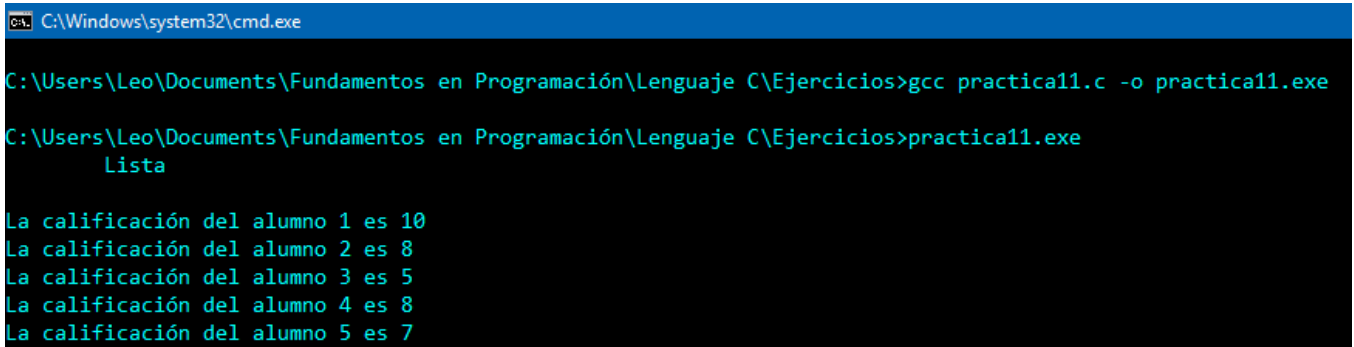
ACTIVIDADES:

Código (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 for.
6   */
7
8  int main()
9  {
10     #define TAMANO 5
11     int lista[TAMANO] = {10, 8, 5, 8, 7};
12     char a = 160, e = 130, i = 161, o = 162, u = 163;
13
14     int indice = 0;
15
16     printf("\tLista\n");
17     while (indice < 5)
18     {
19         printf("\nLa calificación del alumno %d es %d", o, indice+1, lista[indice]);
20         indice += 1; // análogo a indice = indice + 1;
21     }
22
23     printf("\n");
24     return 0;
25 }
```

Compilación (arreglo unidimensional while)



```
C:\Windows\system32\cmd.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>gcc practica11.c -o practica11.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>practica11.exe
Lista

La calificación del alumno 1 es 10
La calificación del alumno 2 es 8
La calificación del alumno 3 es 5
La calificación del alumno 4 es 8
La calificación del alumno 5 es 7
```

Código (arreglo unidimensional for)

```
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios\practica11.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?

area.c x  primercal.c x  ayp.c x  divisible.c x  practica10a.c x  inventario.c x  practica11.c x

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 for.
6   */
7
8  int main()
9  {
10     #define TAMANO 5
11     int lista[TAMANO] = {10, 8, 5, 8, 7};
12     char a = 160, e = 130, i = 161, o = 162, u = 163;
13
14
15     printf("\tLista\n");
16     for (int indice = 0; indice < 5; indice++)
17     {
18         printf("\nLa calificaci%cn del alumno %d es %d", o, indice+1, lista[indice]);
19     }
20
21     printf("\n");
22     return 0;
23 }
24
```

Compilación (arreglo unidimensional for)

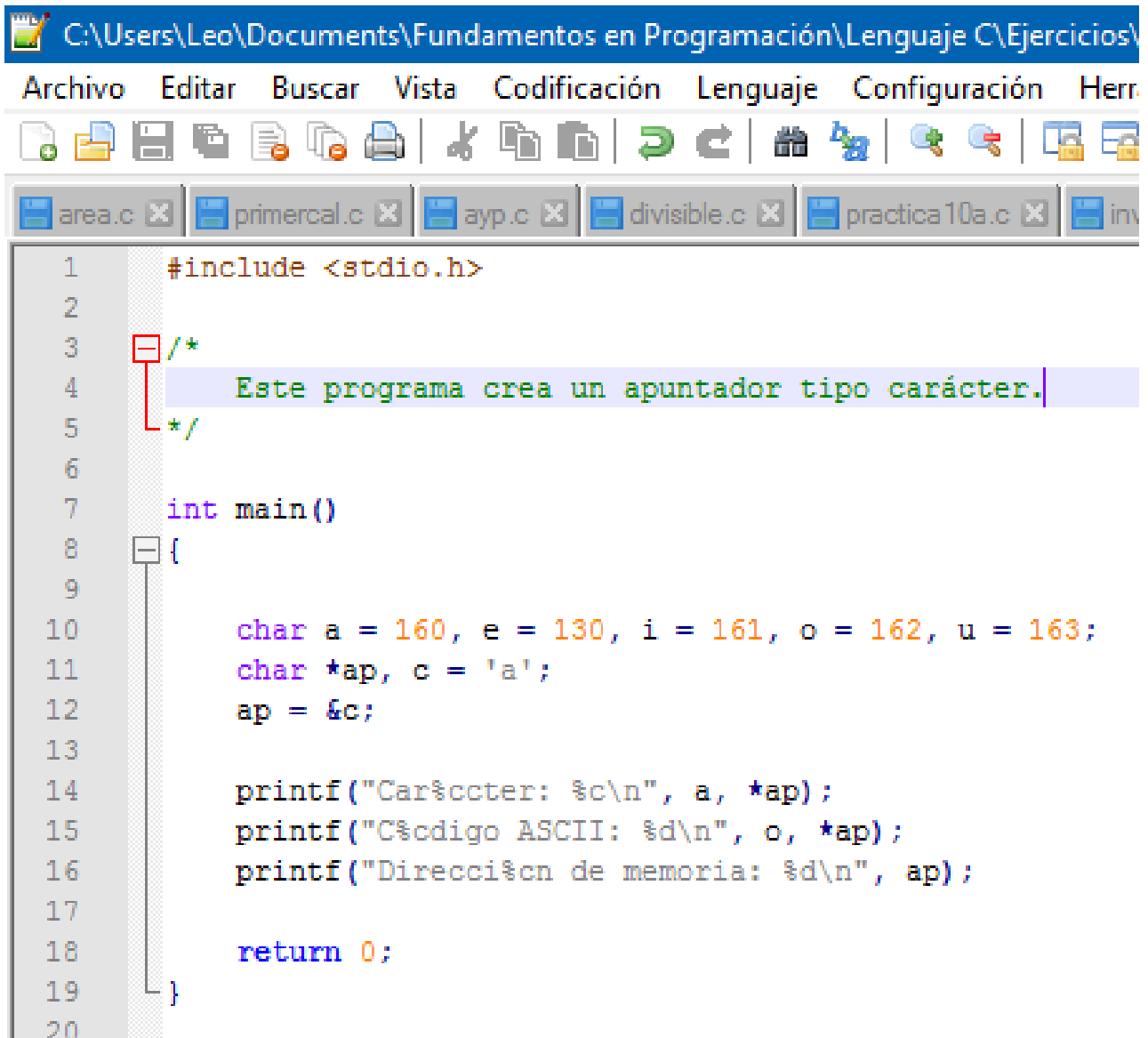
```
C:\> Seleccionar C:\Windows\system32\cmd.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>gcc practica11.c -o practica11.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>practica11.exe
    Lista

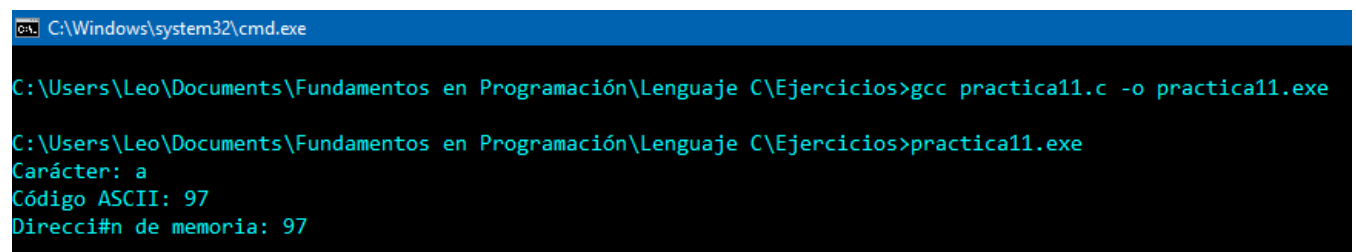
La calificación del alumno 1 es 10
La calificación del alumno 2 es 8
La calificación del alumno 3 es 5
La calificación del alumno 4 es 8
La calificación del alumno 5 es 7
```

Código (apuntadores)



```
1  #include <stdio.h>
2
3  /*
4   Este programa crea un apuntador tipo carácter.
5  */
6
7  int main()
8  {
9
10     char a = 160, e = 130, i = 161, o = 162, u = 163;
11     char *ap, c = 'a';
12     ap = &c;
13
14     printf("Carácter: %c\n", a, *ap);
15     printf("Código ASCII: %d\n", o, *ap);
16     printf("Dirección de memoria: %d\n", ap);
17
18     return 0;
19 }
20
```

Compilación (apuntadores)



```
C:\Windows\system32\cmd.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>gcc practica11.c -o practica11.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>practica11.exe
Carácter: a
Código ASCII: 97
Dirección de memoria: 97
```

Código (apuntadores)

```
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios\practica11.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?

1  #include <stdio.h>
2
3  /*
4   Este programa accede a las localidades de memoria de distintas variables a
5   través de un apuntador.
6  */
7
8  int main()
9  {
10
11     char ac = 160, e = 130, i = 161, o = 162, u = 163;
12     int a = 5, b = 10, c[10] = {5, 4, 3, 1, 9, 8, 7, 6, 0};
13     int *apEnt;
14     apEnt = &a;
15
16     printf("a = 5, b = 10, c [10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}\n");
17     printf("apEnt = %a\n");
18
19     b = *apEnt;
20     printf("b = *apEnt \t-> b = %i\n", b);
21
22     b = *apEnt + 1;
23     printf("b = *apEnt \t-> b = %i\n", b);
24
25     *apEnt = 0;
26     printf("*apEnt = 0 \t-> a = %i\n", a);
27
28     apEnt = &c[0];
29     printf("apEnt = %c[0] \t-> apEnt = %i\n", *apEnt);
30
31
32     return 0;
33 }
34
```

Compilación (apuntadores)

```
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>gcc practica11.c -o practica11.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>practica11.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      -> b = 6
*apEnt = 0      -> a = 0
apEnt = &c[0]   -> apEnt = 5
```

Código (apuntadores)

```
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios\practica11.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  P

area.c x  primercal.c x  ayp.c x  divisible.c x  practica10a.c x  inventario.c x  practica11.c x

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
8  int main()
9  {
10
11     char ac = 160, e = 130, i = 161, o = 162, u = 163;
12     int arr[] = {5, 4, 3, 2, 1};
13     int *apArr;
14     apArr = arr;
15
16     printf("int arr[]={5, 4, 3, 2, 1};\n");
17     printf("apArr = arr[0]\n");
18
19     int x = *apArr;
20     printf("x = * apArr \t -> x = %d\n", x);
21
22     x = *(apArr+1);
23     printf("x = *(apArr+1) \t -> x = %d\n", x);
24
25     x = *(apArr+2);
26     printf("x = *(apArr+1) \t -> x = %d\n", x);
27
28     return 0;
29 }
```

Compilación (apuntadores)

```
C:\Windows\system32\cmd.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>gcc practica11.c -o practica11.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>practica11.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
```

Código (apuntadores en ciclo for)

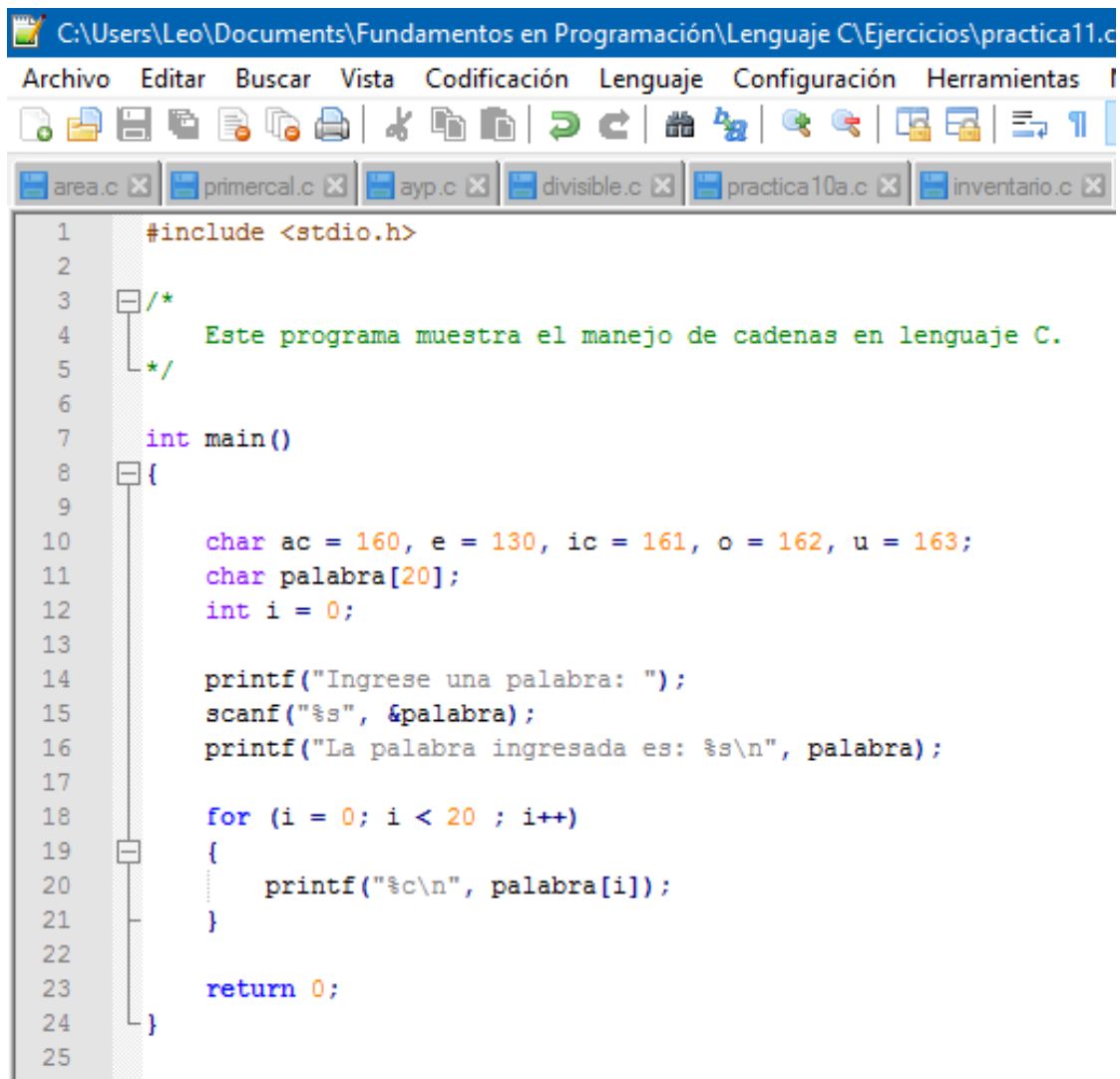
```
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios\practica11.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ver
[area.c] [primercal.c] [ayp.c] [divisible.c] [practica10a.c] [inventario.c] [practica11.c]
1  #include <stdio.h>
2
3  /*
4   * Este programa genera un arreglo unidimensional de 5 elementos y
5   * accede a cada elemento del arreglo a través de un apuntador
6   * utilizando el ciclo for.
7   */
8
9  int main()
10 {
11
12     char ac = 160, e = 130, i = 161, o = 162, u = 163;
13     #define TAMANO 5
14     int lista[TAMANO] = {10, 8, 5, 8, 7};
15     int *ap = lista;
16
17     printf("\tLista\n");
18     for (int indice = 0 ; indice < 5 ; indice++)
19     {
20         printf("\nLa calificaci%cn del alumno %d es %d", o, indice+1, *(ap+indice));
21     }
22     return 0;
23 }
24
```

Compilación (apuntadores en ciclo for)

```
C:\Windows\system32\cmd.exe
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>gcc p
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>pract
Lista

La calificación del alumno 1 es 10
La calificación del alumno 2 es 8
La calificación del alumno 3 es 5
La calificación del alumno 4 es 8
La calificación del alumno 5 es 7
```

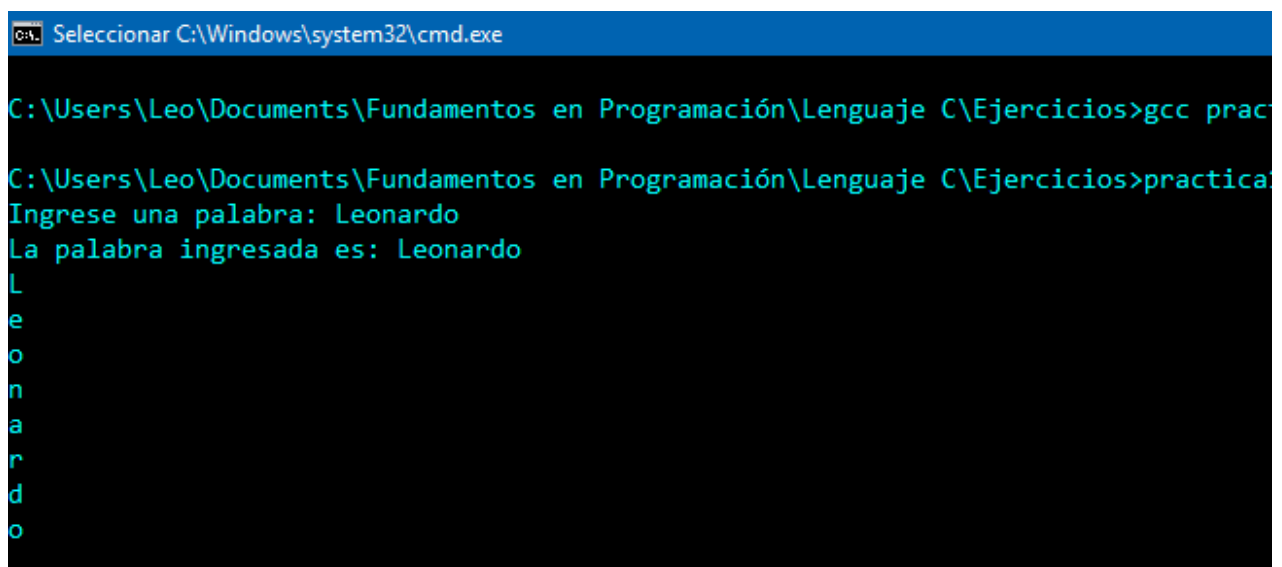

Código (apuntadores en cadenas)



The screenshot shows a C code editor window titled "C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios\practica11.c". The menu bar includes "Archivo", "Editar", "Buscar", "Vista", "Codificación", "Lenguaje", "Configuración", and "Herramientas". The toolbar contains icons for file operations and development tools. The tab bar shows several open files: "area.c", "primercal.c", "ayp.c", "divisible.c", "practica10a.c", and "inventario.c". The main editor area displays the following C code:

```
1  #include <stdio.h>
2
3  /*
4     Este programa muestra el manejo de cadenas en lenguaje C.
5  */
6
7  int main()
8  {
9
10     char ac = 160, e = 130, ic = 161, o = 162, u = 163;
11     char palabra[20];
12     int i = 0;
13
14     printf("Ingrese una palabra: ");
15     scanf("%s", &palabra);
16     printf("La palabra ingresada es: %s\n", palabra);
17
18     for (i = 0; i < 20 ; i++)
19     {
20         printf("%c\n", palabra[i]);
21     }
22
23     return 0;
24 }
25
```

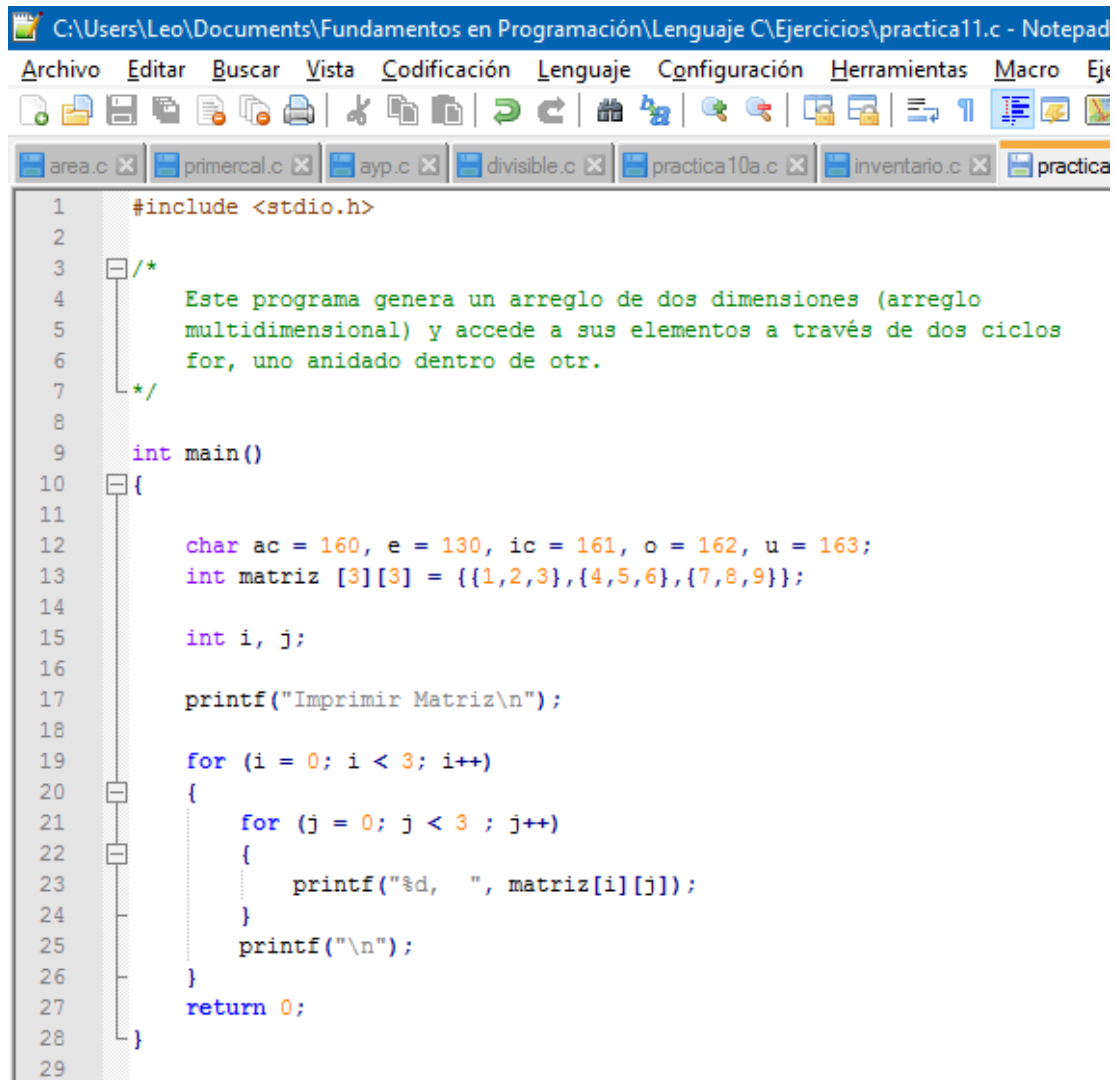
Compilación (apuntadores en cadenas)



The screenshot shows a command prompt window titled "Seleccionar C:\Windows\system32\cmd.exe". The user has navigated to the directory "C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios" and executed the following commands and outputs:

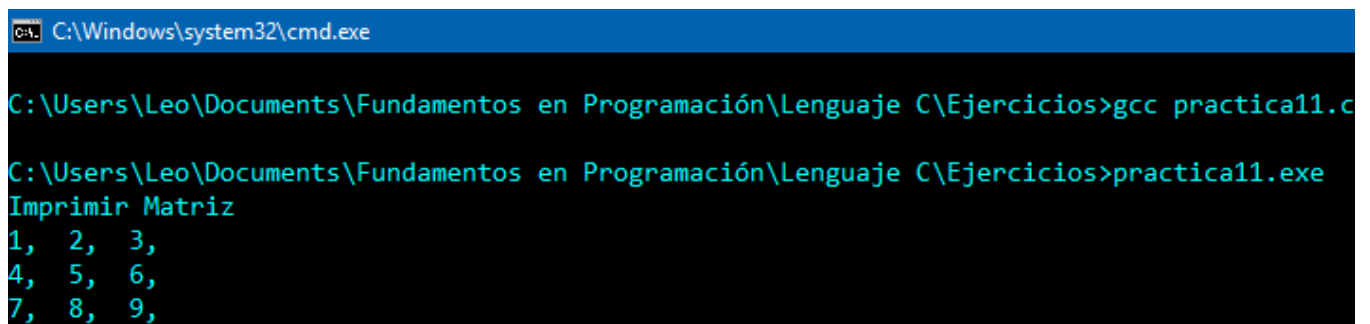
```
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>gcc practica11.c
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>practica11.exe
Ingrese una palabra: Leonardo
La palabra ingresada es: Leonardo
L
e
o
n
a
r
d
o
```

Código (arreglos multidimensionales)



```
1  #include <stdio.h>
2
3  /*
4   Este programa genera un arreglo de dos dimensiones (arreglo
5   multidimensional) y accede a sus elementos a través de dos ciclos
6   for, uno anidado dentro de otr.
7  */
8
9  int main()
10 {
11
12     char ac = 160, e = 130, ic = 161, o = 162, u = 163;
13     int matriz [3][3] = {{1,2,3},{4,5,6},{7,8,9}};
14
15     int i, j;
16
17     printf("Imprimir Matriz\n");
18
19     for (i = 0; i < 3; i++)
20     {
21         for (j = 0; j < 3 ; j++)
22         {
23             printf("%d,  ", matriz[i][j]);
24         }
25         printf("\n");
26     }
27     return 0;
28 }
29
```

Compilación (arreglos multidimensionales)



```
C:\Windows\system32\cmd.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>gcc practica11.c

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>practica11.exe
Imprimir Matriz
1,  2,  3,
4,  5,  6,
7,  8,  9,
```

Código (arreglos multidimensionales con apuntadores)

```
C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios\practica11.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ve
[area.c] [primercal.c] [ayp.c] [divisible.c] [practica10a.c] [inventario.c] [practica11.c]

1  #include <stdio.h>
2
3  /*
4   * Este programa genera un arreglo de dos dimensiones (arreglo
5   * multidimensional) y accede a sus elementos a través de un apuntador utilizando
6   * un ciclo for.
7   */
8
9  int main()
10 {
11
12     char ac = 160, e = 130, ic = 161, o = 162, u = 163;
13
14     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
15
16     int i, cont=0, *ap;
17
18     ap = &matriz[3][3];
19
20     printf("Imprimir Matriz\n");
21     for (i = 0; i < 9; i++)
22     {
23         if (cont == 3)
24         {
25             printf("\n");
26             cont = 0;
27         }
28         printf("%d\t", *(ap+i));
29         cont++;
30     }
31     printf("\n");
32
33     return 0;
34 }
35
```

Código (arreglos multidimensionales con apuntadores)

```
C:\Windows\system32\cmd.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>gcc practica11.c -o practica11.exe

C:\Users\Leo\Documents\Fundamentos en Programación\Lenguaje C\Ejercicios>practica11.exe
Imprimir Matriz
0      1      2
3      4      5
6      7      8
```

CONCLUSIONES:

Se demostró durante todo el desarrollo de la práctica el uso e importancia de los arreglos con diferentes ejemplos y ejercicios de código y compilación que mostraban en la terminal el resultado de almacenar los datos por medio de cada arreglo. Así mismo se vio como los arreglos funcionan además gracias a estructuras de repetición que permiten crear una construcción sistemática en el código para que el proceso de imprimir cada dato del arreglo se repita hasta que en pantalla se muestren todos los elementos que lo componen.

Se comprendió la utilidad de un apuntador dentro del tema de la programación, se vieron ejemplos bastante claros de cómo un arreglo te muestra la ubicación de un dato en particular. Para mí en lo personal me llamo mucho la atención el ejemplo donde usamos el carácter a pues me di cuenta que con la ayuda de un apuntador, se puede encontrar el valor ASCII de un carácter del teclado, es algo que se utiliza mucho y está muy bien conocer esa herramienta que sirve para ello.