	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:	Marco Antonio Martínez Quintana
Asignatura:	Fundamentos de Programación
Grupo:	3
No de Práctica(s):	Práctica 11
Integrante(s):	Santiago Díaz Edwin Jaret
No. de Equipo de cómputo empleado:	No aplica
No. de Lista o Brigada:	50
Semestre:	2021-1
Fecha de entrega:	Miércoles 6 de enero del 2020
Observaciones:	

CALIFICACIÓN: _____

Depuración de Programas

Objetivos.

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

Resultados.

1.-

practica11_1.c > main()

```c
#include <stdio.h>
/*
    Este programa genera un arreglo undimensional de 5 elementos y los
    accede a cada elemento del arreglo a través de un ciclo while.
*/

int main () {
    #define TAMANO 5
    int lista[TAMANO] = {10, 8, 5, 8, 7};

    int indice = 0;

    printf("\t Lista \n");
    while (indice < 5) {
        printf("\n Calificacion del alumno %d es %d", indice+1, lista[indice]);
        indice += 1;   //Análogo a índice = indice + 1;
    }

    printf("\n");

    return 0;

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica11_1.c -o 11
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\11
         Lista

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

2.-

practica11_2.c > main()

```c
#include <stdio.h>
/*
    Este programa genera un arreglo undimensional de 5 elementos y los
    accede a cada elemento del arreglo a través de un ciclo for.
*/

int main () {
    #define TAMANO 5
    int lista[TAMANO] = {10, 8, 5, 8, 7};
    int indice;

    printf("\t Lista \n");

    for (indice = 0; indice < 5; indice++) {
        printf("\n Calificacion del alumno %d es %d", indice+1, lista[indice]);
    }

    printf("\n");

    return 0;

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica11_2.c -o 12
C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\12
   Lista

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

3.-

pratica11_3.c > main()

```c
#include <stdio.h>
/*
    Este programa crea un apuntador de tipo carácter.
*/

int main () {
    char *ap, c = 'a';
    ap = &c;

    printf("Caracter: %c \n", *ap);
    printf("Codigo ASCII: %d\n",*ap);
    printf("Direccion de memoria: %d\n",ap);
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc pratica11_3.c -o 13
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\13
Caracter: a
Codigo ASCII: 97
Direccion de memoria: 6487575
```

4.-

practica11_4.c > main()

```c
#include <stdio.h>
/*
    Este programa accede a las localidades de memoria de distintas variables
    a través de un apuntador.
*/

int main () {
    int a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};
    int *apEnt;
    apEnt = &a;

    printf("A = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 9}\n");
    printf("apEnt = &a\n");

    b = *apEnt;
    printf("b = *apEnt \t -> b = %i\n", b);

    b = *apEnt + 1;
    printf("b = *apEnt + 1 \t -> b = %i\n", b);

    *apEnt = 0;
    printf("*apEnt = 0 \t -> a = %i\n", a);

    apEnt = &c[0];
    printf("apEnt = &c[0] \t -> apEnt = %i \n", *apEnt);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\14
A = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 9}
apEnt = &a
b = *apEnt      -> b = 5
b = *apEnt + 1  -> b = 6
*apEnt = 0      -> a = 0
apEnt = &c[0]   -> apEnt = 5
```

5.-

practica11_5.c > main()

```c
#include <stdio.h>
/*
    Este programa trabaja con aritmética de apuntadores para acceder
    a los valores d eun arreglo.
*/

int main () {
    int arr[] = {5, 4, 3, 2, 1};
    int *apArr;
    apArr = arr;

    printf("int arr[] = {5, 4, 3, 2, 1};\n");
    printf("apArr = &arr[0]\n");

    int x = *apArr;
    printf("x = *apArr \t -> x = %d\n",x);

    x = *(apArr+1);
    printf("x = *(apArr+1) \t -> x = %d\n", x);

    x = *(apArr+2);
    printf("x = *(apArr+1) \t -> x = %d\n", x);
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica11_5.c -o 15
 C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\15
arr[] = {5, 4, 3, 2, 1};
apArr = &arr[0]
x = *apArr       -> x = 5
x = *(apArr+1)   -> x = 4
x = *(apArr+1)   -> x = 3
```

6.-

```c
#include <stdio.h>
/*
 Este programa genera un arreglo unidimensional de 5 elementos y
accede a cada elemento del arreglo a través de un apuntador
utilizando un ciclo for.
*/
int main (){
    #define TAMANO 5
    int lista[TAMANO] = {10, 8, 5, 8, 7};
    int *ap = lista;
    int indice;
    printf("\tLista\n");
    for (indice = 0; indice < 5 ; indice++){
        printf("\nCalificacion del alumno %d es %d", indice+1, *(ap+indice));
    }

    printf("\n");

    return 0;
}
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica11_6.c -o 16
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\16
        Lista

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

7.- A este código le hice unos cambios ya que si no incluía la librería String.h me imprimía la palabra junto con espacios para rellenar el valor de caracteres hasta 20. Por lo tanto, lo modifiqué para que solo se imprima las letras ingresadas.

C practica11_7.c > main()

```c
#include <stdio.h>
#include <string.h>
/*
    Este programa muestra el manejo de cadenas en lenguaje C.
*/
int main(){
    char palabra[20];
    int i=0, num;
    printf("Ingrese una palabra: ");
    scanf("%s", palabra);
    num = strlen(palabra);
    printf("La palabra ingresada es: %s\n", palabra);
    for (i = 0 ; i < num ; i++){
        printf("%c\n", palabra[i]);
    }
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica11_7.c -o 17
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\17
Ingrese una palabra: BocinaNegra
La palabra ingresada es: BocinaNegra
B
o
c
i
n
a
N
e
g
r
a
```

8.-

practica11_8.c > main()

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica11_8.c -o 18
 PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\18
  Imprimir Matriz
1, 2, 3,
4, 5, 6,
7, 8, 9,
```

9.- En este código al parecer había 1 problema, le hacia falta un "*" a la variable matriz. Al ejecutarlo como viene en el manual no compilaba y después de un rato vi el por qué no podía. Creo yo que esta es la solución real.

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

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica11_9.c -o 19
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\19
Imprimir Matriz
1       2       3
4       5       6
7       8       9
```

Y este sería el código si quisiéramos conocer en qué parte de la memoria se encuentran los valores

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

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> gcc practica11_9.c -o 19
PS C:\Users\Gerardo\Desktop\INGENIERIA UNAM\Fundamenos de Programación> .\19
Imprimir Matriz
6487520  1      6487520  2      6487520  3
6487520  4      6487520  5      6487520  6
6487520  7      6487520  8      6487520  9
```

Conclusiones.

En esta práctica se aprendió a usar matrices, encontrar en donde que parte de la memoria se encuentran nuestras variables asignadas. También, se aprendió usar listas, en este caso, en el programa 7 se le pide al usuario que ingrese una palabra y dentro del código se procesa para convertirlo en una lista.