



## 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):*

1: APLICACIONES DE ARREGLOS

*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:*

12 MARZO 2021

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

## PRÁCTICA # 1 APLICACIONES DE ARREGLOS

### Objetivo de Laboratorio.

Utilizar arreglos unidimensionales y multidimensionales para dar solución a problemas computacionales.

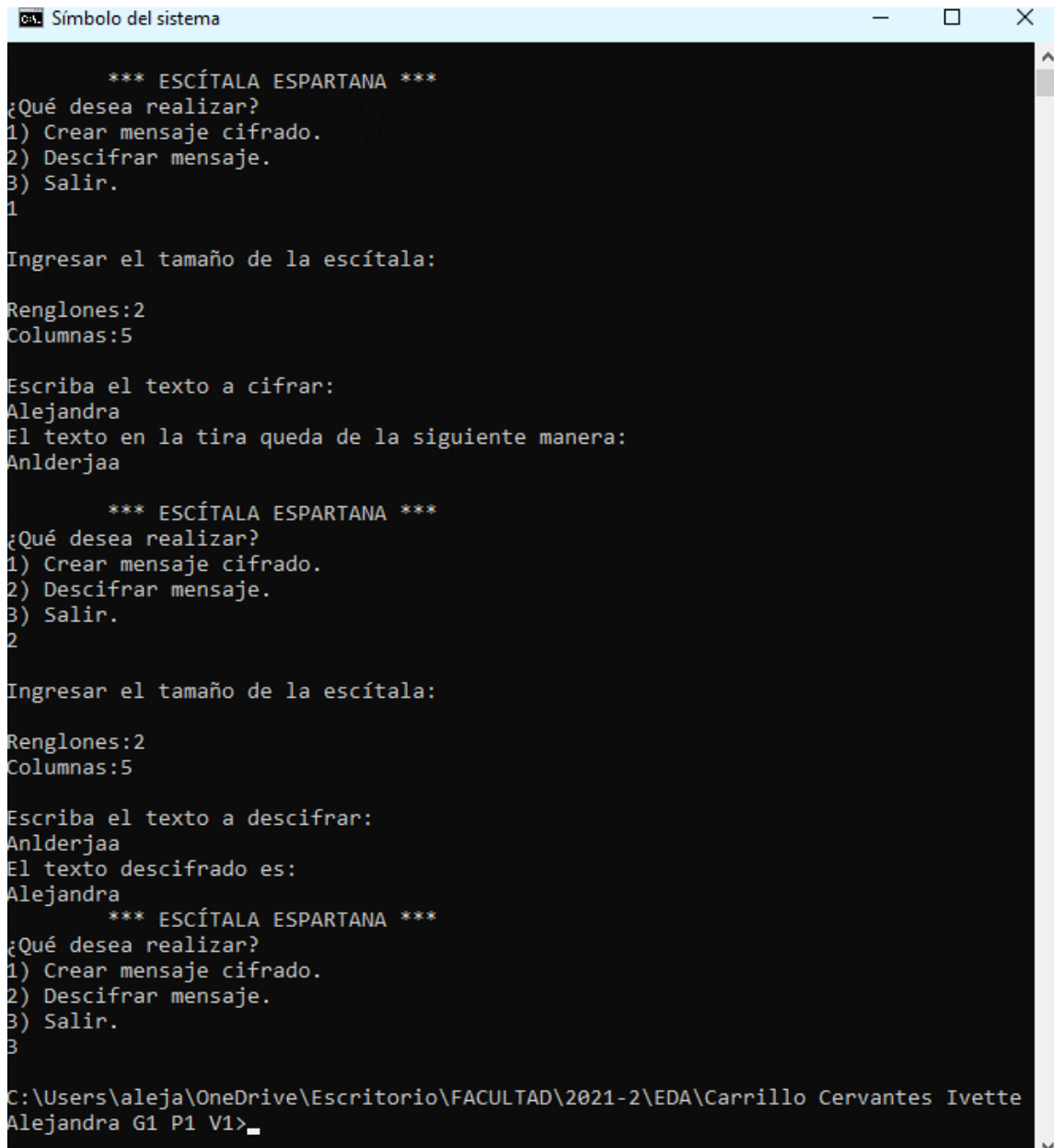
### Objetivo de clase.

El alumno realizará aplicaciones en las que se utilicen arreglos de tipos de datos primitivos. Implementará apropiadamente el manejo de índices en arreglos multidimensionales

### Ejemplos de la guía.

El programa tiene como función, codificar mensajes por el método de Escítala espartana, uno de los primeros métodos criptográficos conocidos proveniente de Esparta. Esto se realiza través de ciclos for y arreglos multidimensionales.

Al momento de ejecutarse, te dará dos opciones, crear o descifrar un mensaje, en la primera opción, te pedirá que escribas un mensaje para que como salida te dé el texto codificado, mientras que al elegir la segunda opción, introduces el mensaje codificado y como salida te muestra el texto original.



```
*** ESCÍTALA ESPARTANA ***
¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
1

Ingresar el tamaño de la escítala:

Renglones:2
Columnas:5

Escriba el texto a cifrar:
Alejandra
El texto en la tira queda de la siguiente manera:
Anlderjaa

*** ESCÍTALA ESPARTANA ***
¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
2

Ingresar el tamaño de la escítala:

Renglones:2
Columnas:5

Escriba el texto a descifrar:
Anlderjaa
El texto descifrado es:
Alejandra

*** ESCÍTALA ESPARTANA ***
¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
3

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Carrillo Cervantes Ivette
Alejandra G1 P1 V1>
```

ejemplo1.c

## EJERCICIOS PROPUESTOS PARA EL DESARROLLO DE LA PRÁCTICA

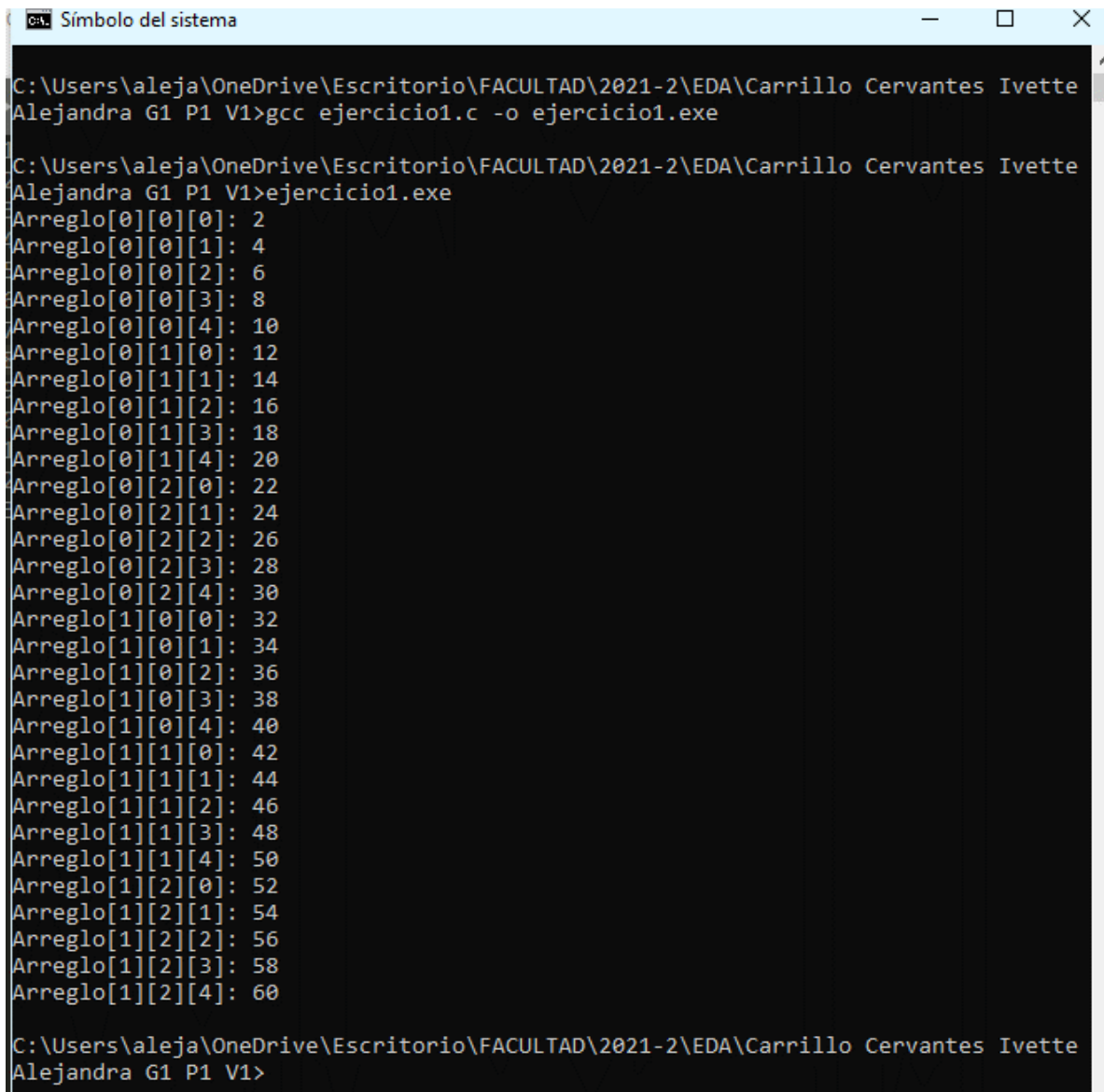
### Ejercicio 1

Codifica, compila y ejecuta el siguiente código, en caso de que se ejecute correctamente indica la salida y qué hace el programa. En caso de que no compile o se obtengan resultados diferentes a los esperados, indica el por qué y realiza la corrección respectiva en el código.

Al momento de compilar el código aparece como salida un error, ya que no aparece los números escritos previamente en el arreglo. Observamos que, en el código, los índices están mal colocados (están al revés en los ciclos for), así que lo corregimos según el orden que vimos en clase y esta vez el programa se ejecuta de manera correcta. Observamos que este tiene como función imprimir los elementos del arreglo antes declarados, indicando en que parte del arreglo se encuentran.

```
6     for (i=0; i<2; i++){
7         for (j=0; j<3; j++){
8             for (k=0; k<5; k++){
9                 printf("Arreglo[%d][%d][%d]: %d\n", i, j, k, arr[i][j][k]);
```

Corrección de código



```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Carrillo Cervantes Ivette
Alejandra G1 P1 V1>gcc ejercicio1.c -o ejercicio1.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Carrillo Cervantes Ivette
Alejandra G1 P1 V1>ejercicio1.exe
Arreglo[0][0][0]: 2
Arreglo[0][0][1]: 4
Arreglo[0][0][2]: 6
Arreglo[0][0][3]: 8
Arreglo[0][0][4]: 10
Arreglo[0][1][0]: 12
Arreglo[0][1][1]: 14
Arreglo[0][1][2]: 16
Arreglo[0][1][3]: 18
Arreglo[0][1][4]: 20
Arreglo[0][2][0]: 22
Arreglo[0][2][1]: 24
Arreglo[0][2][2]: 26
Arreglo[0][2][3]: 28
Arreglo[0][2][4]: 30
Arreglo[1][0][0]: 32
Arreglo[1][0][1]: 34
Arreglo[1][0][2]: 36
Arreglo[1][0][3]: 38
Arreglo[1][0][4]: 40
Arreglo[1][1][0]: 42
Arreglo[1][1][1]: 44
Arreglo[1][1][2]: 46
Arreglo[1][1][3]: 48
Arreglo[1][1][4]: 50
Arreglo[1][2][0]: 52
Arreglo[1][2][1]: 54
Arreglo[1][2][2]: 56
Arreglo[1][2][3]: 58
Arreglo[1][2][4]: 60

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Carrillo Cervantes Ivette
Alejandra G1 P1 V1>
```

*ejercicio1.c*

## Ejercicio 2.

Escribe un programa para trabajar sobre un arreglo de 8 elementos en el cual se mostrará al usuario un menú con las siguientes operaciones:

- 1) Ingresar o modificar elementos del arreglo
- 2) Mostrar la Suma de los elementos
- 3) Realizar la multiplicación de los elementos
- 4) Realizar la suma solo de los elementos divisibles entre 3
- 5) Multiplicar por 3 cada elemento del arreglo

El usuario seleccionará la opción de su preferencia y el programa mostrará los datos del arreglo original y el resultado de la opción seleccionada. El programa debe realizarse en un menú que debe ejecutarse continuamente hasta que el usuario decida salir.

Este programa te da un menú de opciones para elegir que se desea hacer con el arreglo de 8 elementos que, desde un principio debes de ingresar o modificar, en la mayoría de las opciones se ocuparon ciclos for para la repetición de cada uno de los elementos que hay en el arreglo. En este menú, puse una opción más, la cual es “salir”, el menú se estará repitiendo hasta que el usuario presione esta última opción. En cada opción, se realizó una operación distinta; sin embargo, en la opción número 4, se me presentó una dificultad al indicar cuales eran los elementos divisibles entre 3, esto lo solucione con un “if” dentro del ciclo for donde se realizó la suma de cada uno de esos elementos. También en la opción 5 se me volvió a presentar una dificultad, ya que el resultado lo estaba poniendo en una variable y no en otro arreglo, al percatarme de esto lo modifique.

```
Símbolo del sistema
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Carrillo Cervantes Ivette Alejandra G1 P1 V2>gcc Ejercicio2.c -o Ejercicio2.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Carrillo Cervantes Ivette Alejandra G1 P1 V2>Ejercicio2.exe

ARREGLOS

¿Qué desea realizar?
1) Ingresar o modificar elementos del arreglo
2) Mostrar la Suma de los elementos
3) Realizar la multiplicación de los elementos
4) Realizar la suma solo de los elementos divisibles entre 3
5) Multiplicar por 3 cada elemento del arreglo
6) Salir
1
Ingresa Elemento [1]: 4
Ingresa Elemento [2]: 5
Ingresa Elemento [3]: 6
Ingresa Elemento [4]: 7
Ingresa Elemento [5]: 8
Ingresa Elemento [6]: 9
Ingresa Elemento [7]: 10
Ingresa Elemento [8]: 11

¿Qué desea realizar?
1) Ingresar o modificar elementos del arreglo
2) Mostrar la Suma de los elementos
3) Realizar la multiplicación de los elementos
4) Realizar la suma solo de los elementos divisibles entre 3
5) Multiplicar por 3 cada elemento del arreglo
6) Salir
2
Arreglo original: {4,5,6,7,8,9,10,11,}
La suma de los elementos es: 60

¿Qué desea realizar?
```

```

1) Ingresar o modificar elementos del arreglo
2) Mostrar la Suma de los elementos
3) Realizar la multiplicación de los elementos
4) Realizar la suma solo de los elementos divisibles entre 3
5) Multiplicar por 3 cada elemento del arreglo
6) Salir

3

Arreglo original: {4,5,6,7,8,9,10,11,}
La multiplicación de los elementos es: 6652800

¿Qué desea realizar?
1) Ingresar o modificar elementos del arreglo
2) Mostrar la Suma de los elementos
3) Realizar la multiplicación de los elementos
4) Realizar la suma solo de los elementos divisibles entre 3
5) Multiplicar por 3 cada elemento del arreglo
6) Salir

4

Arreglo original: {4,5,6,7,8,9,10,11,}

La suma de los elementos que se pueden dividir entre 3 es: 15

¿Qué desea realizar?
1) Ingresar o modificar elementos del arreglo
2) Mostrar la Suma de los elementos
3) Realizar la multiplicación de los elementos
4) Realizar la suma solo de los elementos divisibles entre 3
5) Multiplicar por 3 cada elemento del arreglo
6) Salir

5

Arreglo original: {4,5,6,7,8,9,10,11,}
El arreglo al multiplicarlo por 3 queda: {12,15,18,21,24,27,30,33,}

¿Qué desea realizar?
1) Ingresar o modificar elementos del arreglo
2) Mostrar la Suma de los elementos
3) Realizar la multiplicación de los elementos
4) Realizar la suma solo de los elementos divisibles entre 3
5) Multiplicar por 3 cada elemento del arreglo
6) Salir

7
Opción no válida.

¿Qué desea realizar?
1) Ingresar o modificar elementos del arreglo
2) Mostrar la Suma de los elementos
3) Realizar la multiplicación de los elementos
4) Realizar la suma solo de los elementos divisibles entre 3
5) Multiplicar por 3 cada elemento del arreglo
6) Salir

6

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Carrillo Cer
vantes Ivette Alejandra G1 P1 V2>_

```

*ejercicio2.c*

### Ejercicio 3. Convirtiendo a Binario

Escribe un programa en el cual se solicite al usuario llenar un arreglo de 9 elementos (únicamente 0's y 1's). Una vez que el usuario llene el arreglo con el número binario, podrá seleccionar entre alguna codificación

a) Binario puro

b) Punto fijo (6 bit entero y 3 bit decimal)

c) Representación de complemento a 2

El programa realizará la conversión y como salida mostrará el resultado en representación decimal.

Este programa al ejecutarlo te pedirá que ingreses un número binario de 9 bits, en caso de ingresar algún otro número que no sea 0 o 1 te mostrará un mensaje diciéndote que no es un número binario.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P1 V2>gcc ejercicio3.c -o ejercicio3.exe

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P1 V2>ejercicio3.exe

      CONVIRTIENDO A BINARIO

Ingresa un número binario de 9 bits
Ingresa bit [1]: 1
Ingresa bit [2]: 4
Ingresa bit [3]: 0
Ingresa bit [4]: 5
Ingresa bit [5]: 1
Ingresa bit [6]: 0
Ingresa bit [7]: 1
Ingresa bit [8]: 0
Ingresa bit [9]: 1
El número no es un binario

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P1 V2>
```

*ejercicio3.c*

En caso de ingresar un número binario, te mostrará un menú en el cual se debe de elegir una opción, este menú se repetirá hasta que el usuario presione salir.

En la primera opción, te da como salida el valor decimal del número binario ingresado, esto se hizo con un for que contaba cada índice del arreglo, desde el ultimo ingresado hasta el primero, y después se elevaba 2 a la potencia donde estuviera un número 1 y al final se sumaban estos resultados. En la segunda opción se me presento una gran dificultad, ya que para dividir al arreglo en dos partes (6 bits para enteros y 3 para decimales) estaba poniendo todo en un ciclo for, pero al recordar la opción anterior, me di cuenta que debía de hacer dos ciclos, uno para enteros donde la condición sería ( $i = 5$ ;  $i \geq 0$ ;  $i--$ ) y haría lo mismo que la opción anterior y otro sería ( $i = 6$ ;  $i < 9$ ;  $i++$ ) iría contando desde el primer bit hasta el último de este ciclo, y después como lo vimos en clase, para el punto fijo, al convertir los decimales, se debía de elevar 2 a la potencia -posición (del lugar donde esté en número 1), finalmente se sumaron los resultados obtenidos de ambos ciclos y te da como salida el numero decimal del punto fijo. Finalmente, en la ultima opción para que la salida diera el complemento a2, primero tuve que hacer el complemento a1 y después utilizar un ciclo de repetición para sumar uno a este complemento a1.

```
Símbolo del sistema
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P1 V2>gcc ejercicio3.c -o ejercicio3.exe
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P1 V2>ejercicio3.exe

      CONVIRTIENDO A BINARIO

Ingresa un número binario de 9 bits
Ingresa bit [1]: 1
Ingresa bit [2]: 0
Ingresa bit [3]: 0
Ingresa bit [4]: 1
Ingresa bit [5]: 0
Ingresa bit [6]: 1
Ingresa bit [7]: 0
Ingresa bit [8]: 0
Ingresa bit [9]: 1

Seleccione una codificación
1) Binario Puro
2) Punto Fijo (6 bit entero y 3 bit decimal)
3) Representación de complemento a2
4) Salir
1

Número Binario: 0 0 0 1 0 1 0 0 1
El valor en decimal 41

Seleccione una codificación
1) Binario Puro
2) Punto Fijo (6 bit entero y 3 bit decimal)
3) Representación de complemento a2
4) Salir
2

Número Binario: 0 0 0 1 0 1 0 0 1
El resultado es: 5.125000

Seleccione una codificación
1) Binario Puro
2) Punto Fijo (6 bit entero y 3 bit decimal)
3) Representación de complemento a2
4) Salir
3

Número Binario: 0 0 0 1 0 1 0 0 1
Complemento a1 : 1 1 1 0 1 0 1 1 0
Complemento a2 : 1 1 1 0 1 0 1 1 1

Seleccione una codificación
1) Binario Puro
2) Punto Fijo (6 bit entero y 3 bit decimal)
3) Representación de complemento a2
4) Salir
5

Opción no válida.

Seleccione una codificación
1) Binario Puro
2) Punto Fijo (6 bit entero y 3 bit decimal)
3) Representación de complemento a2
4) Salir
4

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes Ivette Alejandra G1 P1 V2>
```

*ejercicio3.c*

## **Conclusiones**

Se cumplieron los objetivos tanto de laboratorio como de clase, ya que en esta práctica se utilizaron los arreglos con el fin de entender las aplicaciones que tienen estos para dar solución a distintos tipos problemas y facilitar la codificación de los programas, puesto que con estos es más sencillo almacenar o manipular un conjunto de elementos.

Se logró cubrir el 100% de los ejercicios solicitados, cada uno de estos ejercicios tiene relación con el tema visto en clase acerca de arreglos como se explico en cada uno de los ejercicios. Personalmente esta práctica me llevo más de lo esperado ya que no tenia conocimientos previos de algunas cosas y tuve que investigar, me fallaron muchas veces los códigos al ejecutarlos, pero poco a poco me fui dando cuenta de mis errores y los corregía conforme se presentaban.

Considero que los ejercicios solicitados estuvieron muy bien para esta práctica, se vio el uso de arreglos y como manipularlos de diferentes maneras como se vio en el ejercicio 2 y 3.

## **Referencias**

- Joyanes. Programación en C. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software. 2da Edición
- Galaviz Casas. Sistema Numérico y Representación de Números en Computadora. Facultad de Ciencias. UNAM