	<b>Carátula para entrega de prácticas</b>	
Facultad de Ingeniería	Laboratorio de docencia	

# Laboratorios de computación salas A y B

<i>Profesor:</i>	Alejandro Esteban Pimentel Alarcón
<i>Asignatura:</i>	Fundamentos de Programación
<i>Grupo:</i>	Grupo #3
<i>No de Práctica(s):</i>	Práctica No.11
<i>Integrante(s):</i>	Gómez Matías Paola Donaji / Laureano González David / Ortiz Luciano Gerson Gael
<i>No. de Equipo de cómputo empleado:</i>	
<i>No. de Lista o Brigada:</i>	No. de Lista (Por orden de Nombre): 17 / 25 / 38 No. de Cuenta (Por orden de nombre): #9472 / #9519 / #1949
<i>Semestre:</i>	2020-1
<i>Fecha de entrega:</i>	28 de Octubre de 2019
<i>Observaciones:</i>	Tarde entrega. Y omitieron el código en la actividad 1

# Arreglos unidimensionales y multidimensionales

Los arreglos, son un conjunto de valores que se relacionan por un nombre en común. Se puede decir también, que son datos que se almacenan bajo el mismo nombre y se diferencian a través de un índice

## Arreglos unidimensionales

Un arreglo unidimensional es una lista de valores guardados bajo el mismo nombre y mismo tipo. Todos los valores dentro de este se les conoce como elementos del arreglo

Para declarar un arreglo unidimensional se hace de la siguiente manera:

**“Tipo de dato” “identificador”[“tamaño”]**

El tipo de dato se introduce el tipo de dato que queremos (Entero, caracter, flotante, etc), en identificador se introduce como llamaremos al arreglo para identificarlo más fácil., y en los corchetes va el tamaño del arreglo

## Arreglos multidimensionales

Los arreglos multidimensionales tienen más de una dimensión, Las dimensiones se manejan por medio de un par de corchetes, dentro de los que se escriben los valores de cada dimensión, separados por comas.

## Objetivo

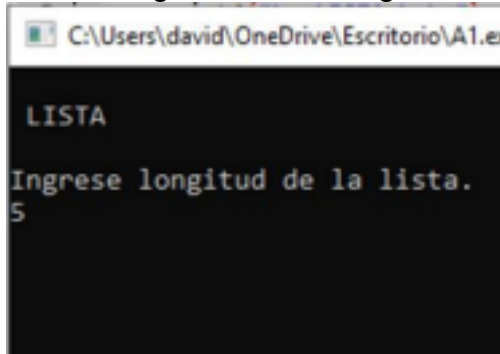
- Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

# Actividades

- Actividad No. 1

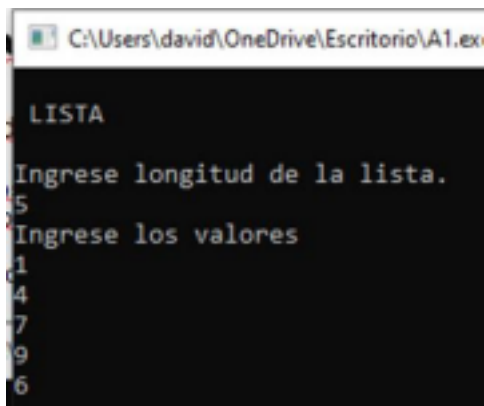
Hicimos un programa el cual pide un número, hace un arreglo de ese tamaño, pide al usuario los números necesarios para llenar el arreglo y muestra el mayor y el menor

Primero ingresamos la longitud del arreglo



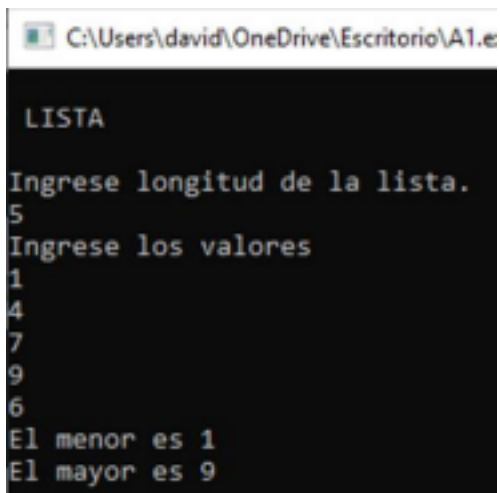
```
C:\Users\david\OneDrive\Escritorio\A1.e
LISTA
Ingrese longitud de la lista.
5
```

Después ingresamos los valores del arreglo



```
C:\Users\david\OneDrive\Escritorio\A1.e
LISTA
Ingrese longitud de la lista.
5
Ingrese los valores
1
4
7
9
6
```

Y al final nos muestra el valor mayor y el valor menor



```
C:\Users\david\OneDrive\Escritorio\A1.e
LISTA
Ingrese longitud de la lista.
5
Ingrese los valores
1
4
7
9
6
El menor es 1
El mayor es 9
```

Falta el código

- Actividad No. 2

Hicimos un programa que pide un número, genera dos matrices del tamaño del producto del número por si mismo, después pide números suficientes para llenar las matrices y muestre al usuario la matriz resultado

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  void suma2matrices(int N){
4
5      int m[N][N];
6      int n[N][N];
7      int suma[N][N];
8      int row;
9      int col;
10     int idmatriz;
11     char pausa;
12     for (row=0;row<N;row++){
13         for (col=0;col<N;col++){
14             m[row][col]=0;
15             n[row][col]=0;
16             suma[row][col]=0;
17         }
18     }
19     for(idmatriz=1;idmatriz<=2;idmatriz++){
20         for (row=0;row<N;row++){
21             for (col=0;col<N;col++){
22                 if(idmatriz==1){
23                     printf("\n valor de casilla [%d][%d] en matriz m: ",row,col);
24                     fflush( stdin );
25                     scanf( "%d", &m[row][col]);
26                 }
27                 else{
28                     printf("\n valor de casilla [%d][%d] en matriz n: ",row,col);
29                     fflush( stdin );
30                     scanf( "%d", &n[row][col]);
31                 }
32             }
33         }
34     }
35 }

```

```

36     for (row=0;row<N;row++){
37         for (col=0;col<N;col++){
38             suma[row][col]=m[row][col] + n[row][col];
39         }
40     }
41     printf("matriz m\n");
42     for (row=0;row<N;row++){
43         for (col=0;col<N;col++){
44             printf("%d ",m[row][col]);
45         }
46         printf("\n");
47     }
48     printf("\n");
49     printf("matriz n\n");
50     for (row=0;row<N;row++){
51         for (col=0;col<N;col++){
52             printf("%d ",m[row][col]);
53         }
54         printf("\n");
55     }
56     printf("\n");
57     printf("matriz suma\n");
58     for (row=0;row<N;row++){
59         for (col=0;col<N;col++){
60             printf("%d ",suma[row][col]);
61         }
62         printf("\n");
63     }
64     printf("Copie los resultados y presione una tecla para continuar");
65     fflush( stdin );
66     pausa = getchar();
67 }

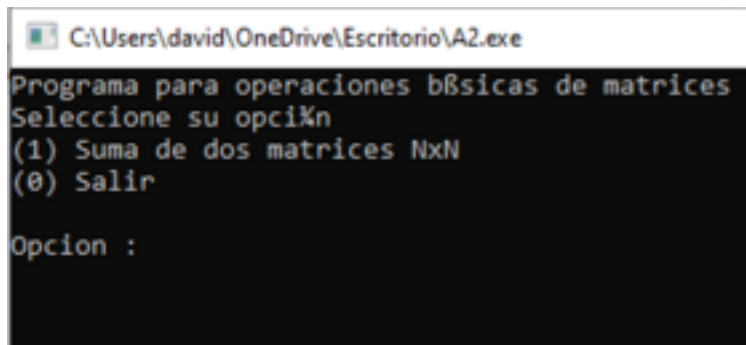
```

```

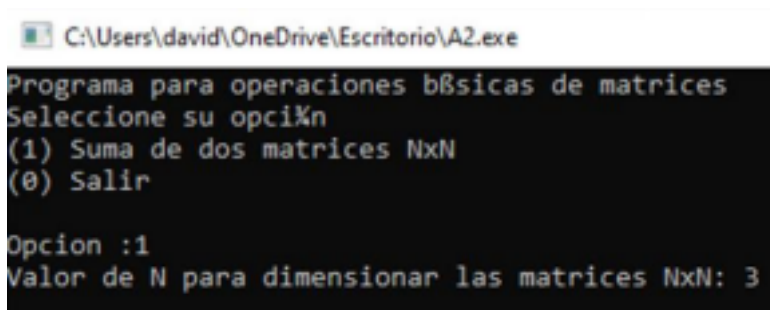
71 void menu()
72 {
73     int iopc;
74     int Naux;
75     do{
76         printf("Programa para operaciones básicas de matrices\n");
77         printf("Seleccione su opción\n");
78         printf("(1) Suma de dos matrices NxN\n");
79         printf("(0) Salir\n\n");
80
81         printf("Opcion :");
82         fflush( stdin );
83         scanf( "%d", &iopc );
84         if (iopc !=0){
85             printf("Valor de N para dimensionar las matrices NxN: ");
86             fflush( stdin );
87             scanf( "%d", &Naux );
88
89             switch ( iopc )
90             {
91                 case 1: suma2matrices(Naux);
92                     break;
93                 default:
94                     printf( "\n Adios" );
95                     break;
96             }
97         }
98     }while( iopc != 0 );
99 }
100
101
102 int main()
103 {
104     menu();

```

Primero seleccionamos una opción de lo que vamos a hacer, en este caso 1, para hacer la matriz NxN



Después elegimos el tamaño de la matriz



Después nos pedirá valores para llenar las matrices y empezamos a elegir valores

```
C:\Users\david\OneDrive\Escritorio\A2.exe
Programa para operaciones básicas de matrices
seleccione su opción
(1) Suma de dos matrices NxN
(0) Salir

Opción :1
Valor de N para dimensionar las matrices NxN: 3

valor de casilla [0][0] en matriz m: 1
valor de casilla [0][1] en matriz m: 2
valor de casilla [0][2] en matriz m: 3
valor de casilla [1][0] en matriz m: 6
valor de casilla [1][1] en matriz m: 8
valor de casilla [1][2] en matriz m: 2
valor de casilla [2][0] en matriz m: 3
valor de casilla [2][1] en matriz m: 4
valor de casilla [2][2] en matriz m: 9
valor de casilla [0][0] en matriz n: 1
valor de casilla [0][1] en matriz n: 2
```

Una vez llenas las dos matrices, nos las muestra, y después nos muestra la matriz suma resultado

```
C:\Users\david\OneDrive\Escritorio\A2.exe
valor de casilla [0][1] en matriz n: 2
valor de casilla [0][2] en matriz n: 5
valor de casilla [1][0] en matriz n: 6
valor de casilla [1][1] en matriz n: 7
valor de casilla [1][2] en matriz n: 1
valor de casilla [2][0] en matriz n: 2
valor de casilla [2][1] en matriz n: 3
valor de casilla [2][2] en matriz n: 4
matriz m
1 2 3
6 8 2
3 4 9
matriz n
1 2 3
6 8 2
3 4 9
matriz suma
2 4 8
12 15 3
5 7 13
```

Al final nos pedirá elegir nuevamente una opción, 1 para hacer una matriz o 0 para salir, en este caso elegimos 0 para terminar el proceso

# Conclusión

Gracias a los arreglos podemos tener diferentes valores organizados en una lista, y con estos arreglos podemos realizar diversas opciones con ellos, como lo hicimos en la segunda actividad sumando dos arreglos multidimensionales llamados matrices, para que nos diera la matriz resultado de la suma.