



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* ALEJANDRO ESTEBAN PIMENTEL ALARCÓN

*Asignatura:* FUNDAMENTOS DE PROGRAMACIÓN

*Grupo:* 3

*No de Práctica(s):* 11

*Integrante(s):* Carlos Sotelo Leyva  
Alan Cárdenas Belmont Gerardo Gutiérrez Soto

*No. de Equipo de  
cómputo empleado:* 38

*No. de Lista o Brigada:* #2751  
#5783 #9267

*Semestre:* 2020-1

*Fecha de entrega:* 28/10/2019

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## INTRODUCCIÓN:

Los arreglos unidimensionales, son estructuras de datos típicamente estáticas en la mayoría de los lenguajes de programación, usan posiciones de memoria que están contiguas y que se indexan de forma numérica. Un arreglo unidimensional es una lista de valores guardados bajo el mismo nombre y del mismo tipo. Cada valor dentro del arreglo se le conoce como elemento del arreglo.

En C los arreglos se declaran de la siguiente forma:

TipoDeDato NombreDelArreglo[TamañoDelArreglo];

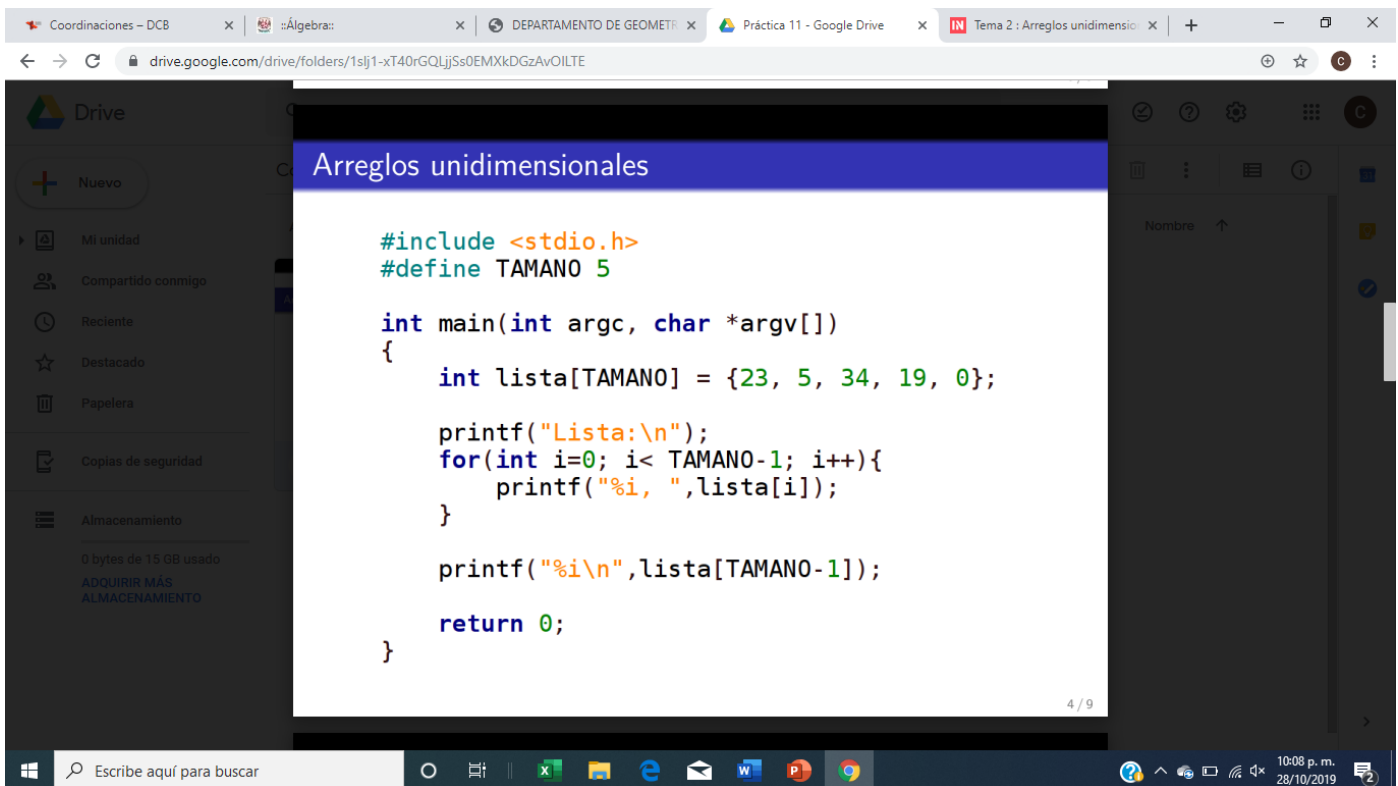
Como podemos ver, es similar a declarar una variable convencional, solo que se coloca entre corchetes el número de posiciones del arreglo, por lo que todas las posiciones serán del mismo tipo. Para llenar todos los elementos del arreglo es común emplear un ciclo que nos permita recorrer el arreglo desde la primera hasta la última posición.

Objetivo de la práctica:

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.

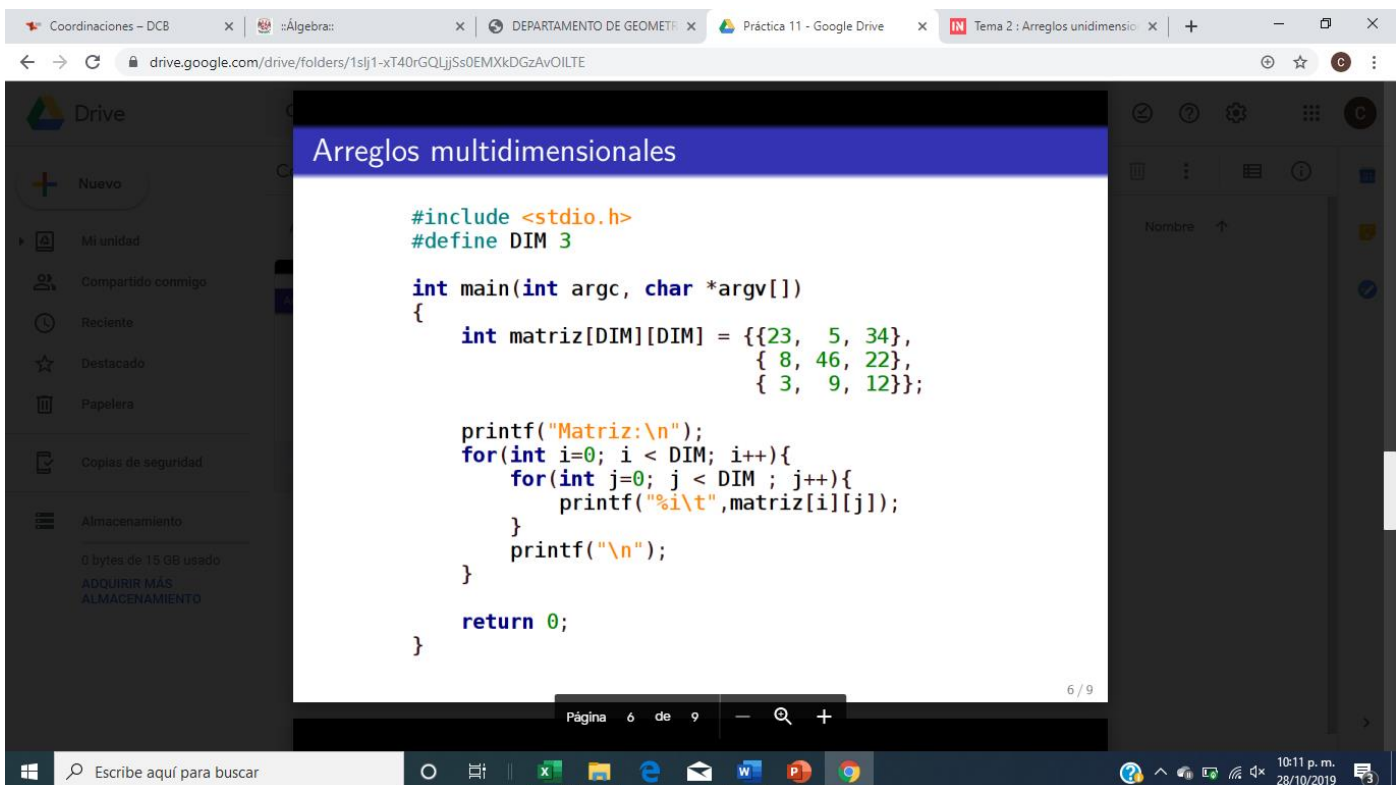
## ARREGLOS UNIDIMENSIONALES:

En esta práctica veremos los arreglos unidimensionales los cuales nos sirven para poner varios valores dentro de una lista, en la cual nos diga el tamaño o de que tan extensa sea esa lista.



## ARREGLOS MULTIDIMENSIONALES:

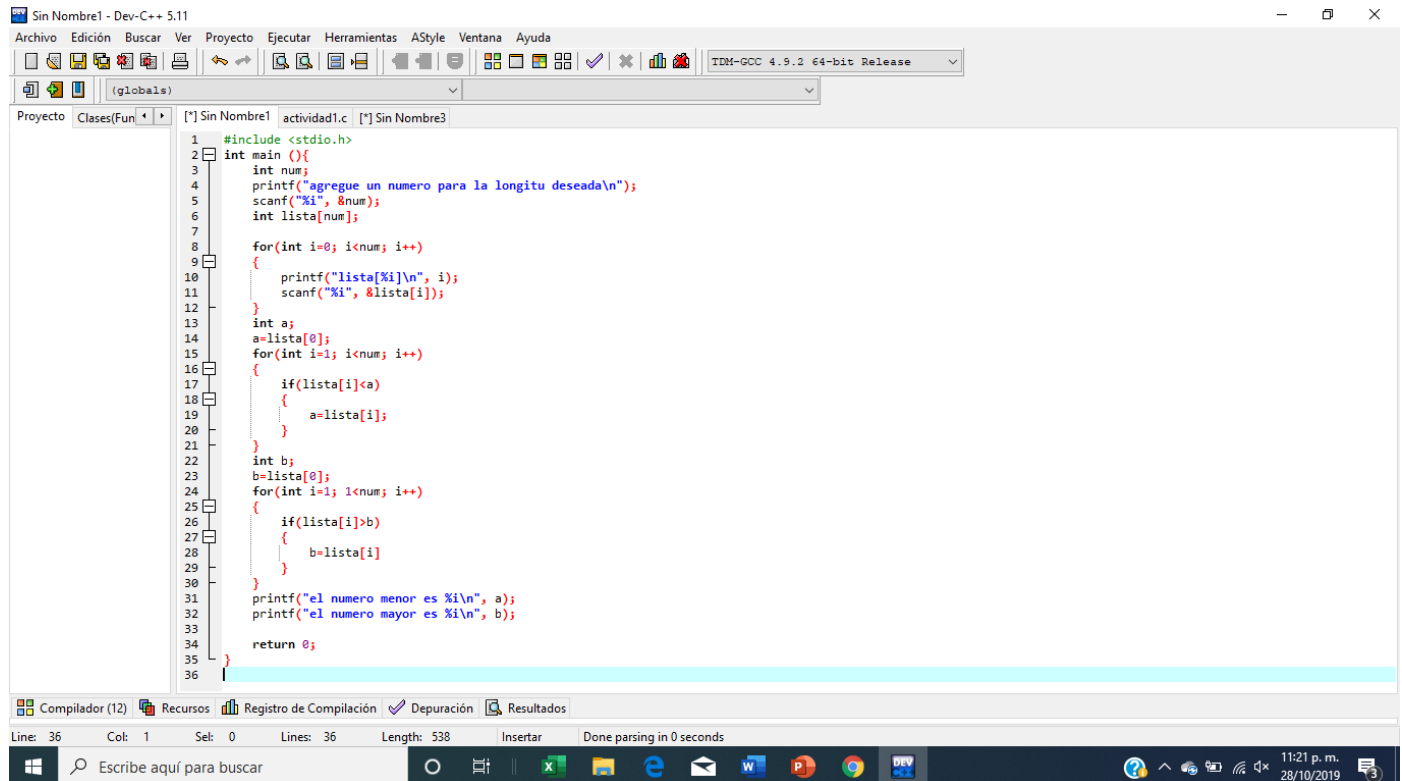
Como su nombre lo dice, en este tipo de arreglos Podemos meter varias dimensiones o listas dentro de un mismo programa poniendo las variables correspondientes a cada uno.



## ACTIVIDAD 1:

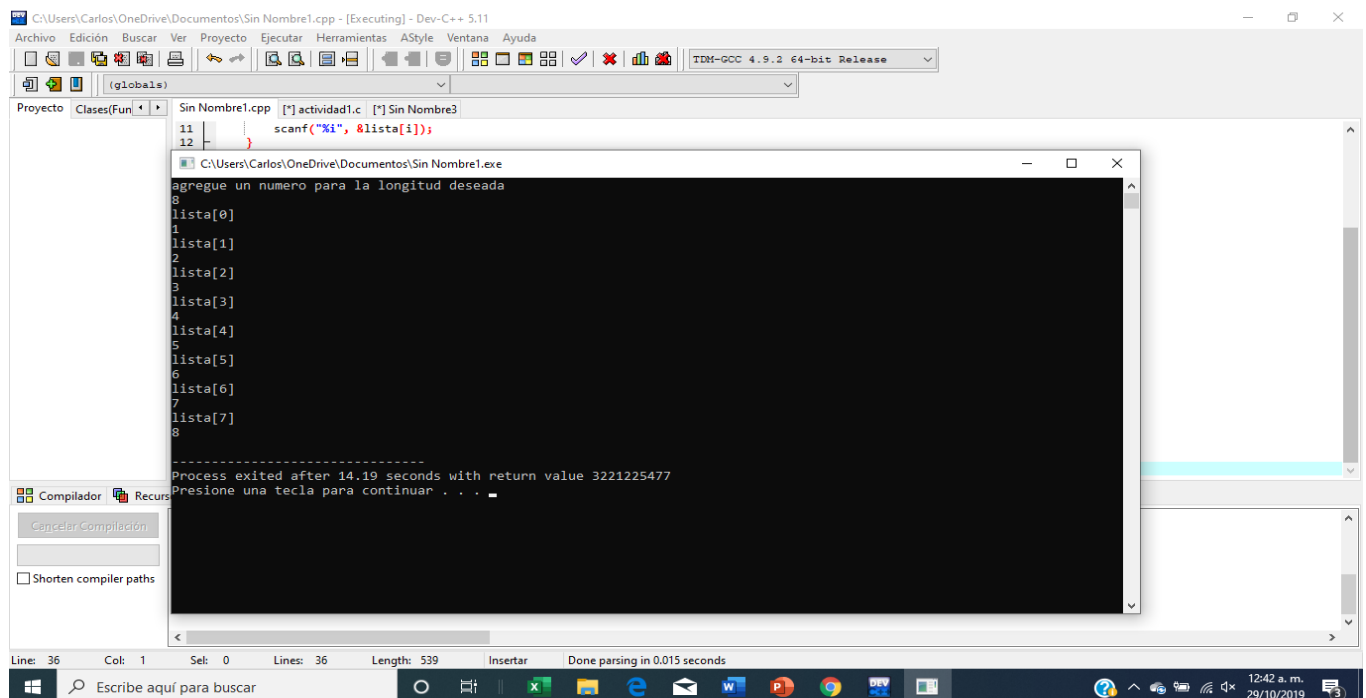
Hacer un programa que:

- Pida al usuario un número.
- Genere un arreglo de esa longitud.
- Pida al usuario números suficientes para llenar el arreglo.
- Muestre al usuario el número menor y el mayor de dicho arreglo.



The screenshot shows the Dev-C++ IDE with the file 'actividad1.c' open. The code is as follows:

```
1 #include <stdio.h>
2 int main () {
3     int num;
4     printf("agregue un numero para la longitud deseada\n");
5     scanf("%i", &num);
6     int lista[num];
7
8     for(int i=0; i<num; i++)
9     {
10        printf("lista[%i]\n", i);
11        scanf("%i", &lista[i]);
12    }
13
14    int a;
15    a=lista[0];
16    for(int i=1; i<num; i++)
17    {
18        if(lista[i]<a)
19        {
20            a=lista[i];
21        }
22    }
23    int b;
24    b=lista[0];
25    for(int i=1; i<num; i++)
26    {
27        if(lista[i]>b)
28        {
29            b=lista[i];
30        }
31    }
32    printf("el numero menor es %i\n", a);
33    printf("el numero mayor es %i\n", b);
34
35    return 0;
36 }
```



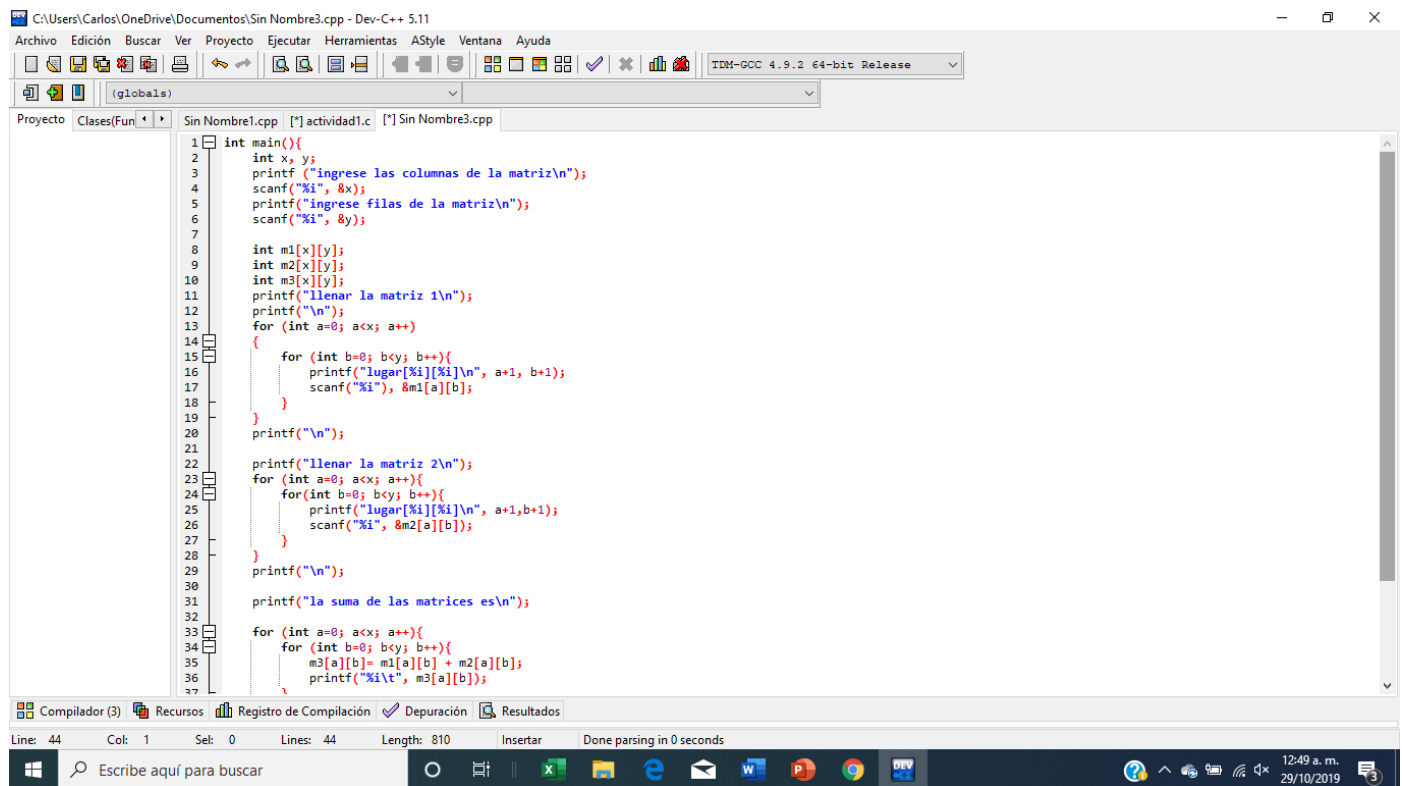
The screenshot shows the Dev-C++ IDE with the program being executed. A console window is open, displaying the following output:

```
agregue un numero para la longitud deseada
8
lista[0]
1
lista[1]
2
lista[2]
3
lista[3]
4
lista[4]
5
lista[5]
6
lista[6]
7
lista[7]
8
-----
Process exited after 14.19 seconds with return value 3221225477
Presione una tecla para continuar . . .
```

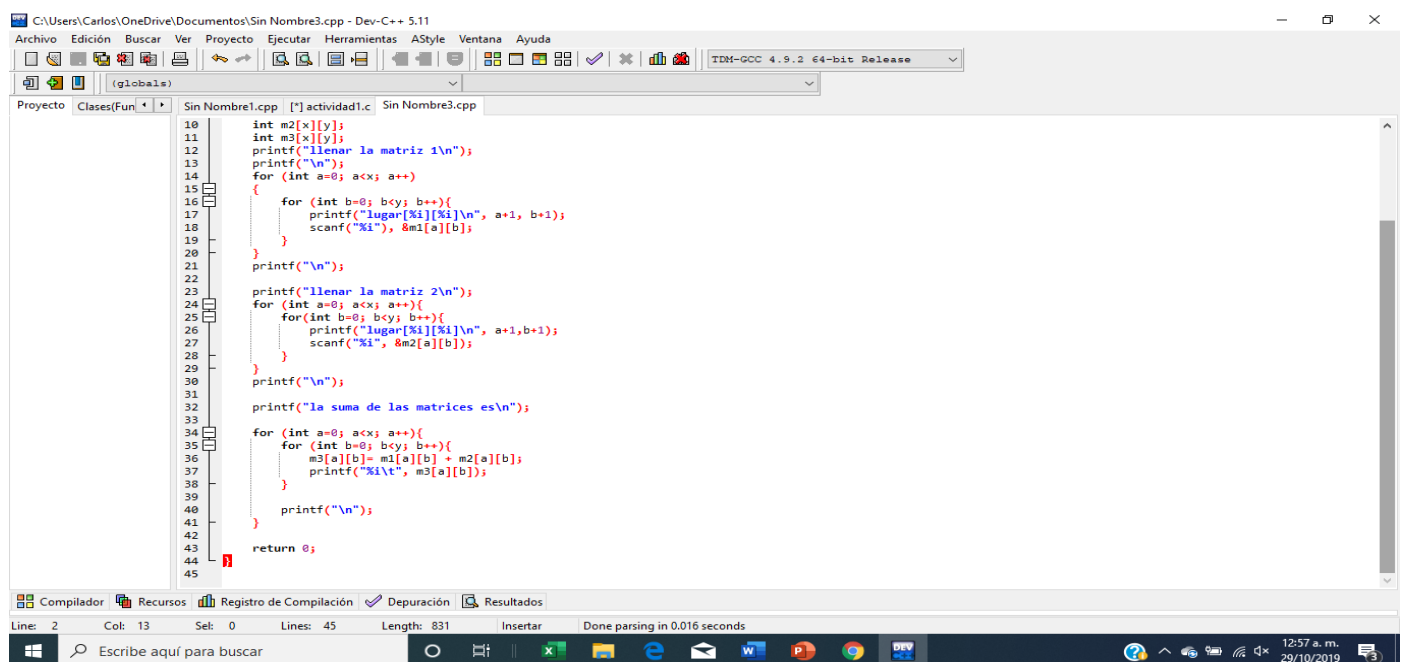
## ACTIVIDAD 2:

Hacer un programa que:

- Pida al usuario un dos números N y M.
- Genere dos matrices de N x M.
- Pida al usuario números suficientes para llenar ambas matrices.
- Muestre al usuario la matriz resultado de sumar las dos de entrada.



```
1 int main(){
2     int x, y;
3     printf("ingrese las columnas de la matriz\n");
4     scanf("%i", &x);
5     printf("ingrese filas de la matriz\n");
6     scanf("%i", &y);
7
8     int m1[x][y];
9     int m2[x][y];
10    int m3[x][y];
11    printf("llenar la matriz 1\n");
12    printf("\n");
13    for (int a=0; a<x; a++){
14        for (int b=0; b<y; b++){
15            printf("lugar[%i][%i]\n", a+1, b+1);
16            scanf("%i", &m1[a][b]);
17        }
18    }
19    printf("\n");
20
21    printf("llenar la matriz 2\n");
22    for (int a=0; a<x; a++){
23        for (int b=0; b<y; b++){
24            printf("lugar[%i][%i]\n", a+1, b+1);
25            scanf("%i", &m2[a][b]);
26        }
27    }
28    printf("\n");
29    printf("la suma de las matrices es\n");
30
31    for (int a=0; a<x; a++){
32        for (int b=0; b<y; b++){
33            m3[a][b] = m1[a][b] + m2[a][b];
34            printf("%i\t", m3[a][b]);
35        }
36    }
37 }
```



```
10 int m2[x][y];
11 int m3[x][y];
12 printf("llenar la matriz 1\n");
13 printf("\n");
14 for (int a=0; a<x; a++){
15     for (int b=0; b<y; b++){
16         printf("lugar[%i][%i]\n", a+1, b+1);
17         scanf("%i", &m1[a][b]);
18     }
19 }
20 printf("\n");
21 printf("llenar la matriz 2\n");
22 for (int a=0; a<x; a++){
23     for (int b=0; b<y; b++){
24         printf("lugar[%i][%i]\n", a+1, b+1);
25         scanf("%i", &m2[a][b]);
26     }
27 }
28 printf("\n");
29 printf("la suma de las matrices es\n");
30
31 for (int a=0; a<x; a++){
32     for (int b=0; b<y; b++){
33         m3[a][b] = m1[a][b] + m2[a][b];
34         printf("%i\t", m3[a][b]);
35     }
36 }
37 printf("\n");
38
39 return 0;
40
41
42
43
44
45 }
```

The screenshot shows a Dev-C++ window titled "C:\Users\Carlos\OneDrive\Documentos\Sin Nombre3.cpp - [Executing] - Dev-C++ 5.11". The main window displays the output of a C++ program. The program initializes two 2x3 matrices, 'lugar' and 'lugar2', with values ranging from 10 to 58. It then prints the sum of these matrices. The output is as follows:

```
lugar[1][3]
3
lugar[2][1]
10
lugar[2][2]
12
lugar[2][3]
15
llenar la matriz 2
lugar[1][1]
11
lugar[1][2]
45
lugar[1][3]
89
lugar[2][1]
85
lugar[2][2]
36
lugar[2][3]
58
la suma de las matrices es
4214829 45      6487505
85      -401057436 32825
-----
Process exited after 39.35 seconds with return value 0
Presione una tecla para continuar . . .
```

Below the output window, the "Compilation results..." panel shows the following information:

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Carlos\OneDrive\Documentos\Sin Nombre3.exe
- Output Size: 131.4423828125 KiB
- Compilation Time: 0.31s

The bottom status bar indicates the current line and column: "Line: 2 Col: 13 Sel: 0 Lines: 45 Length: 831 Insertar Done parsing in 0.016 seconds". The Windows taskbar at the bottom shows the time as 12:56 a.m. on 29/10/2019.

## Reporte de la práctica:

En esta práctica empleamos los arreglos unidimensionales y los arreglos multidimensionales, ya que para cada una de las actividades nos pedían programas con los diferentes arreglos, y se pudieron realizar a base de los conocimientos aprendidos en clases y practicas anteriores, lo nuevo de esta práctica son los arreglos ya que nosotros no habíamos escuchado sobre esto nunca, y hemos visto que son de gran utilidad para hacer operaciones mas largas simplemente utilizando un arreglo ya se unidimensional o multidimensional.