|  |  |
| --- | --- |
|  | **Carátula para entrega de prácticas** |
| Facultad de Ingeniería | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Claudia Rodríguez. |
| *Asignatura:* | Fundamentos de programación. |
| *Grupo:* | 1104 |
| *No de Práctica(s):* | 11 |
| *Integrante(s):* | Álvarez Solano Daira Aketzalli. |
| *Semestre:* | Primero. |
| *Fecha de entrega:* | 29 de octubre del 2018. |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Práctica 11. Arreglos unidimensionales y multidimensionales.**

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.

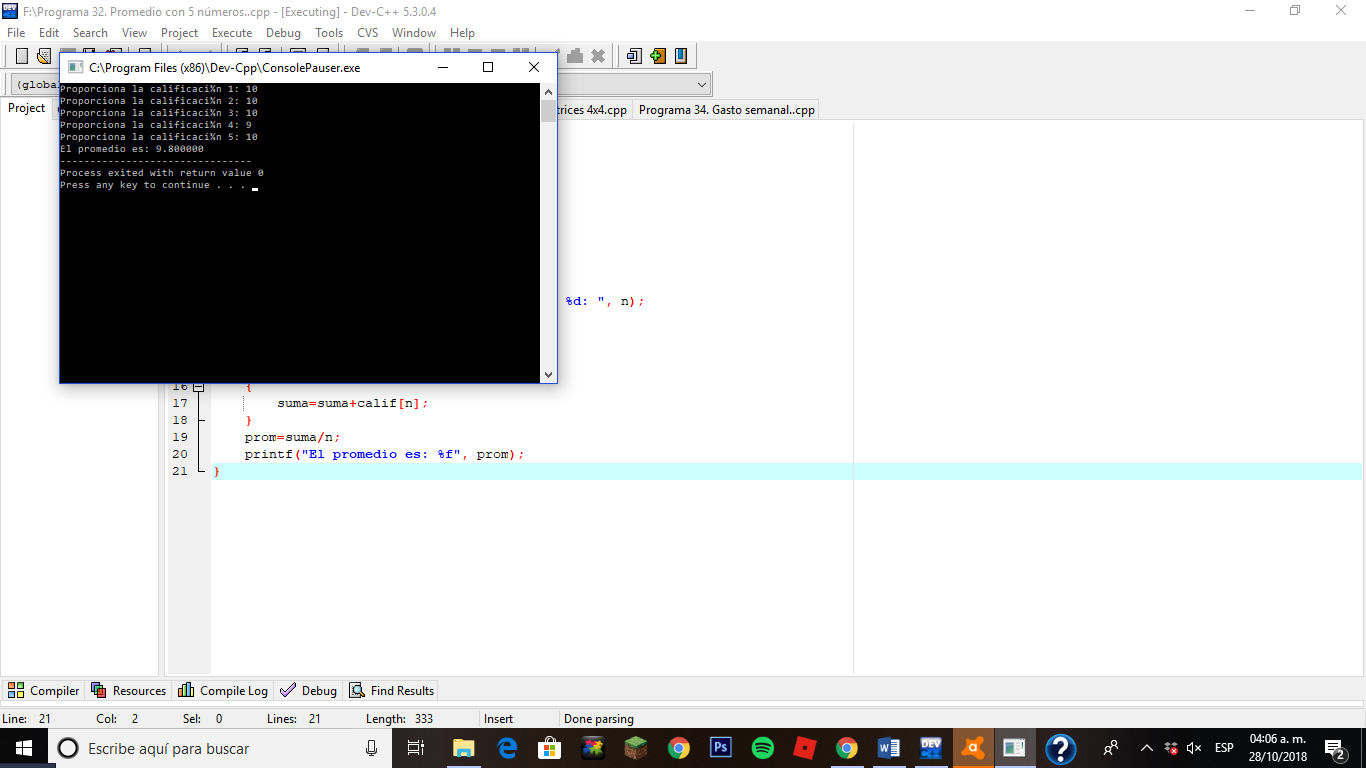
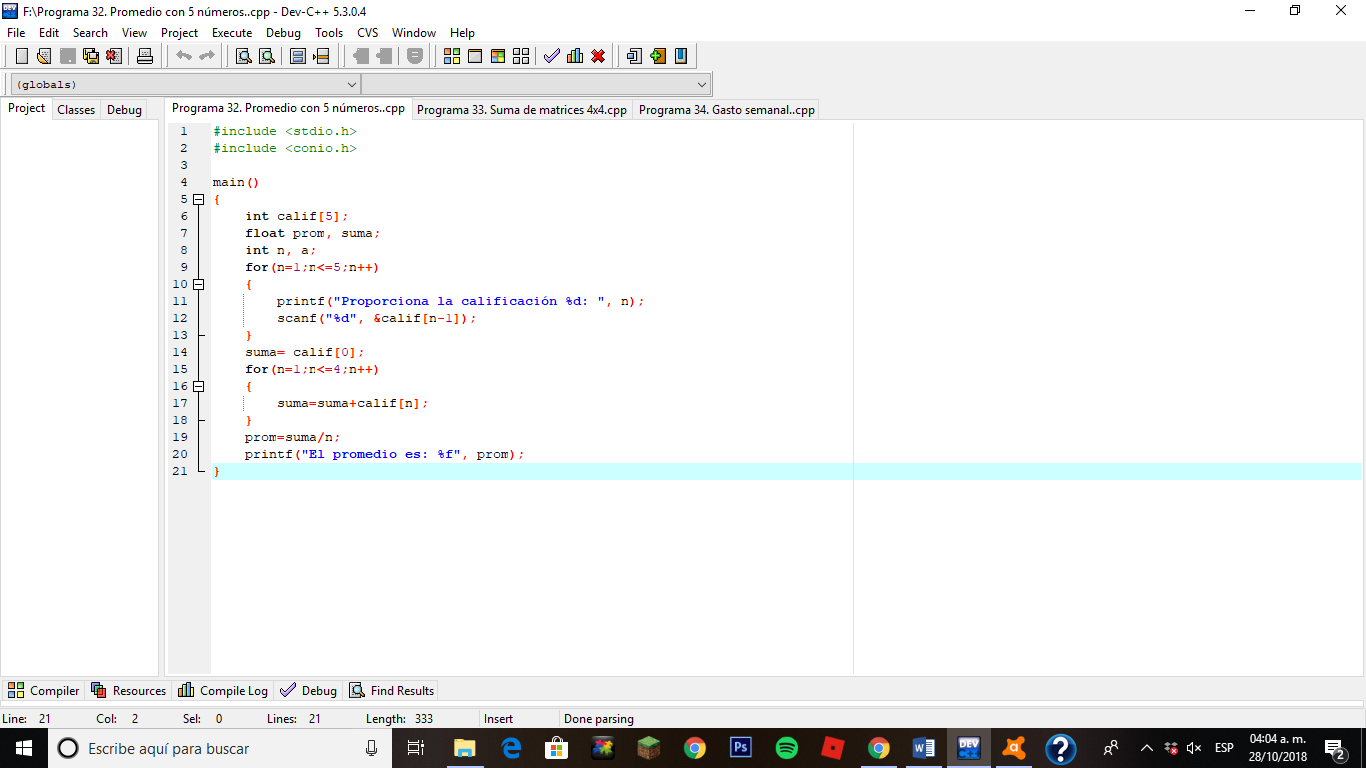
Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse.

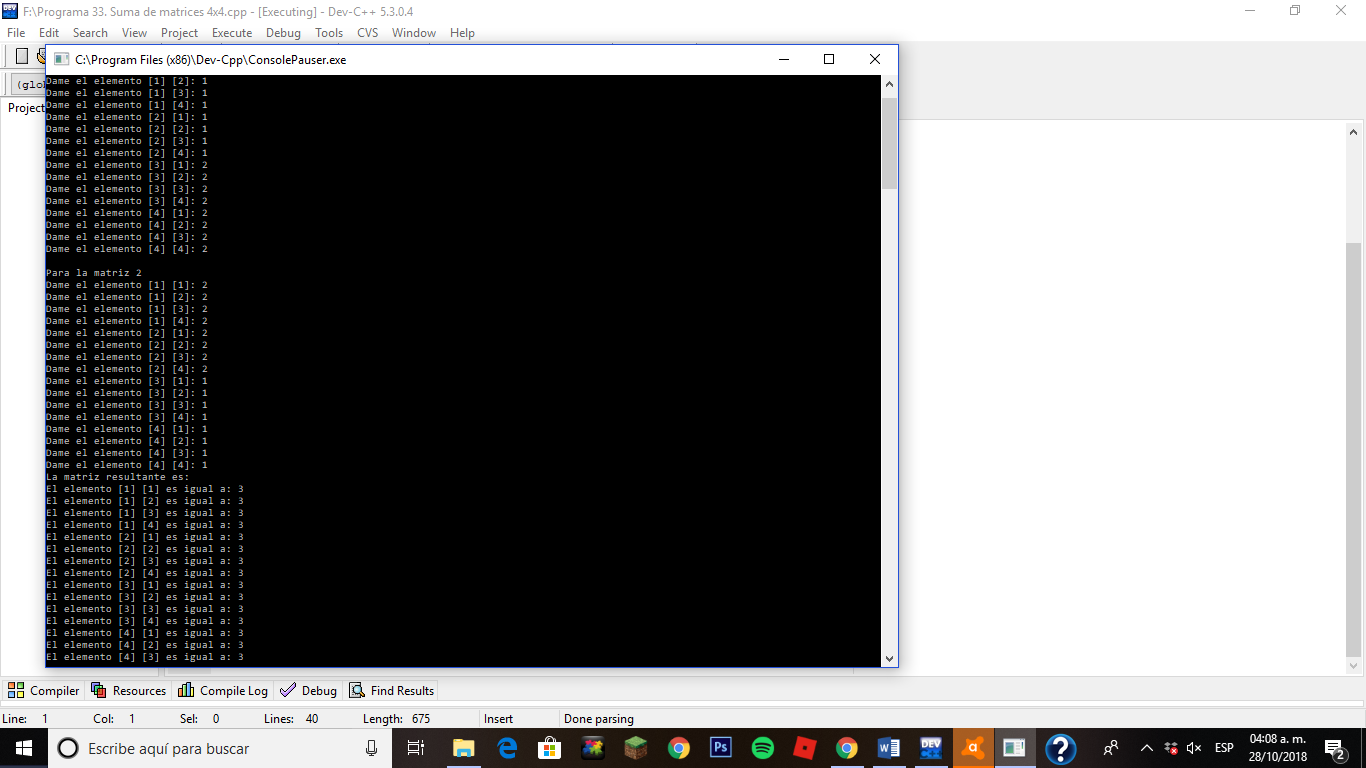
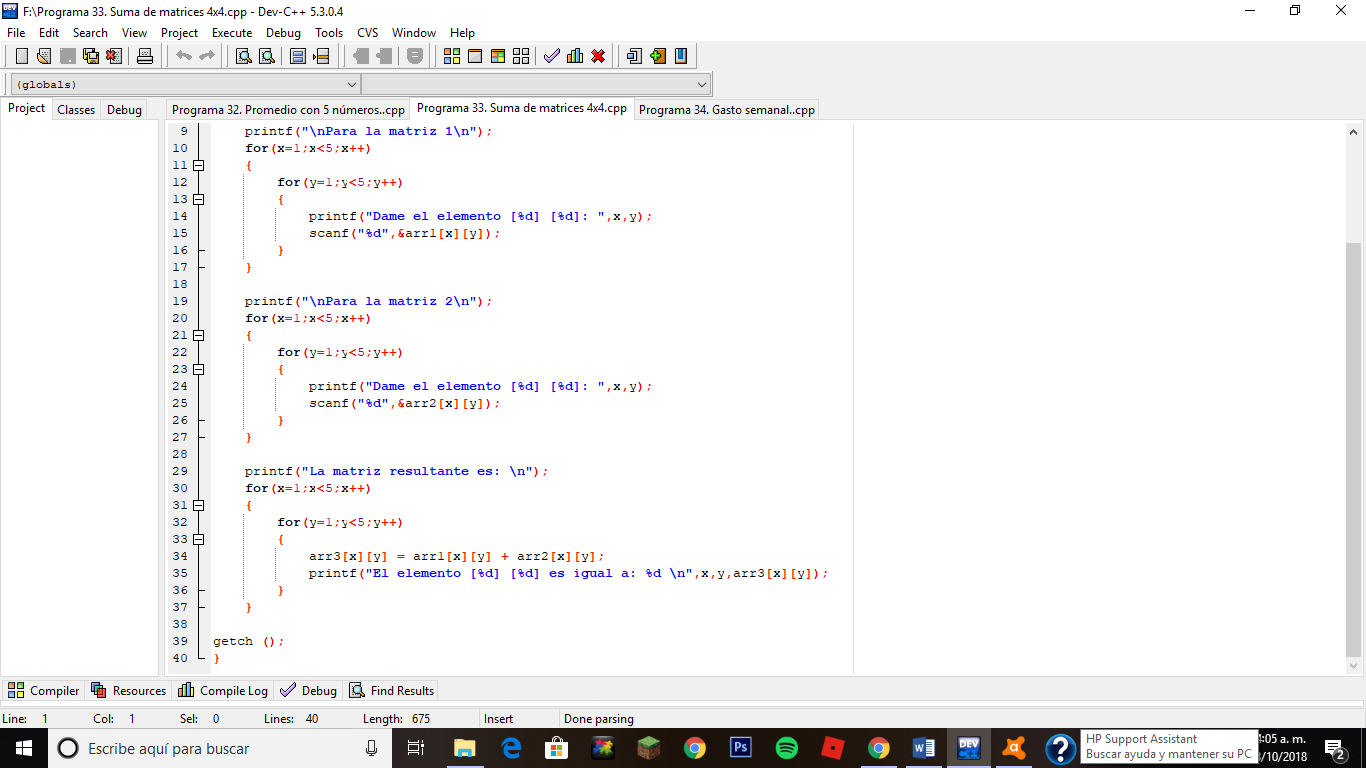
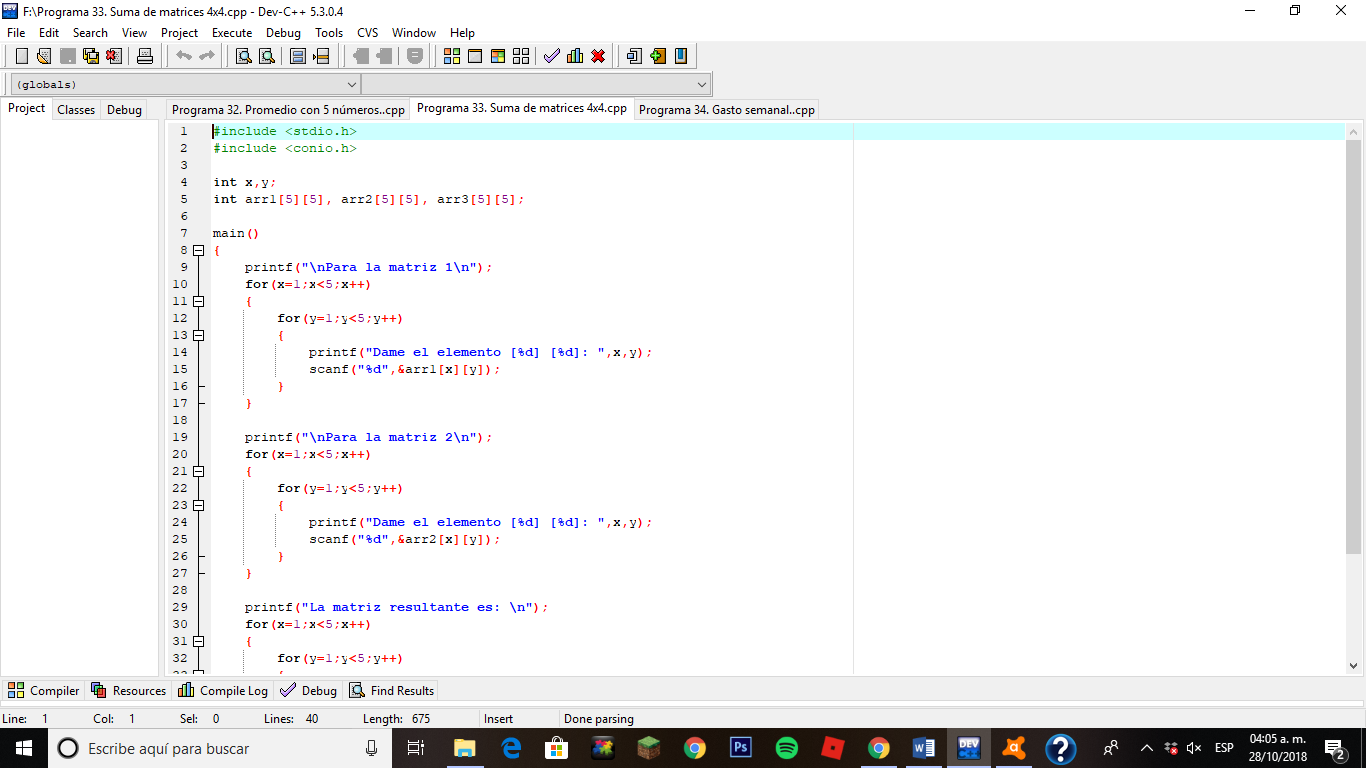
A cada elemento (dato) del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices.

Los arreglos pueden ser unidimensionales o multidimensionales. Los arreglos se utilizan para hacer más eficiente el código de un programa.

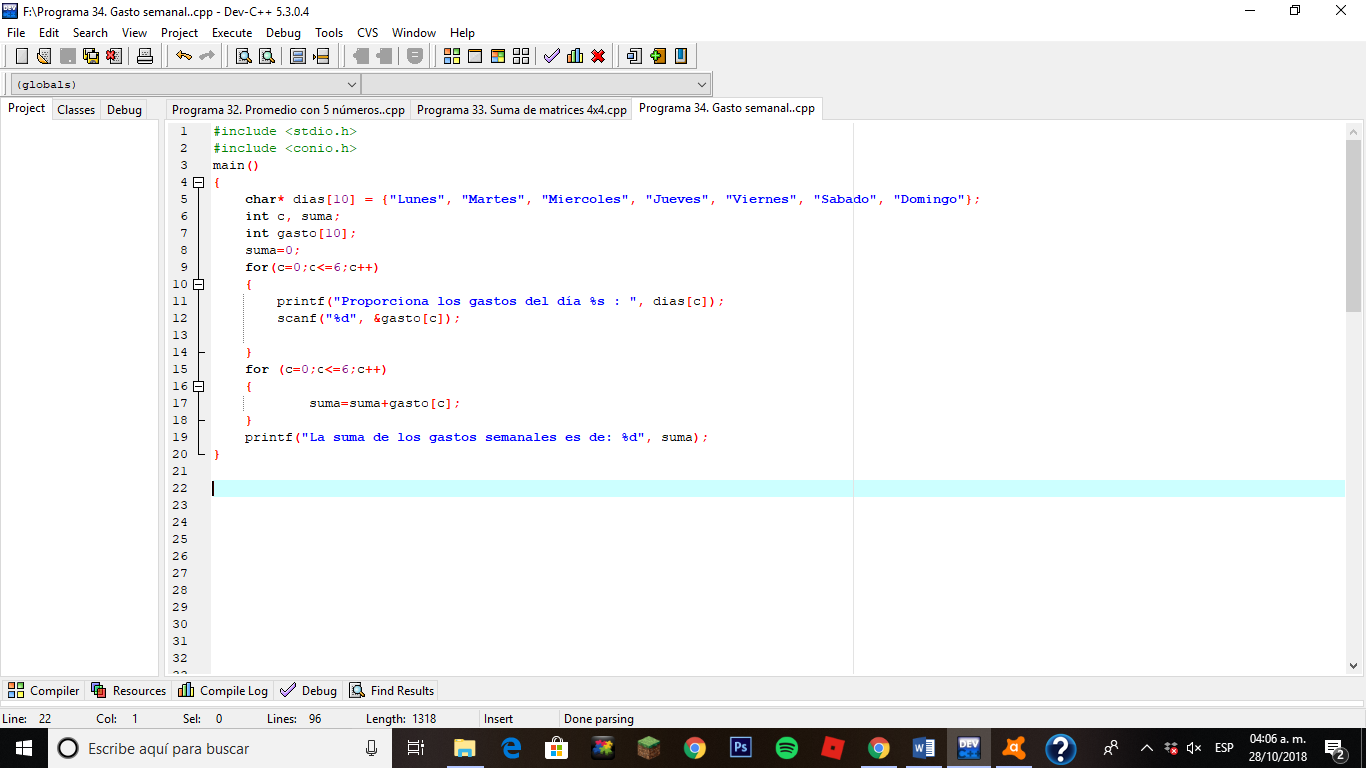
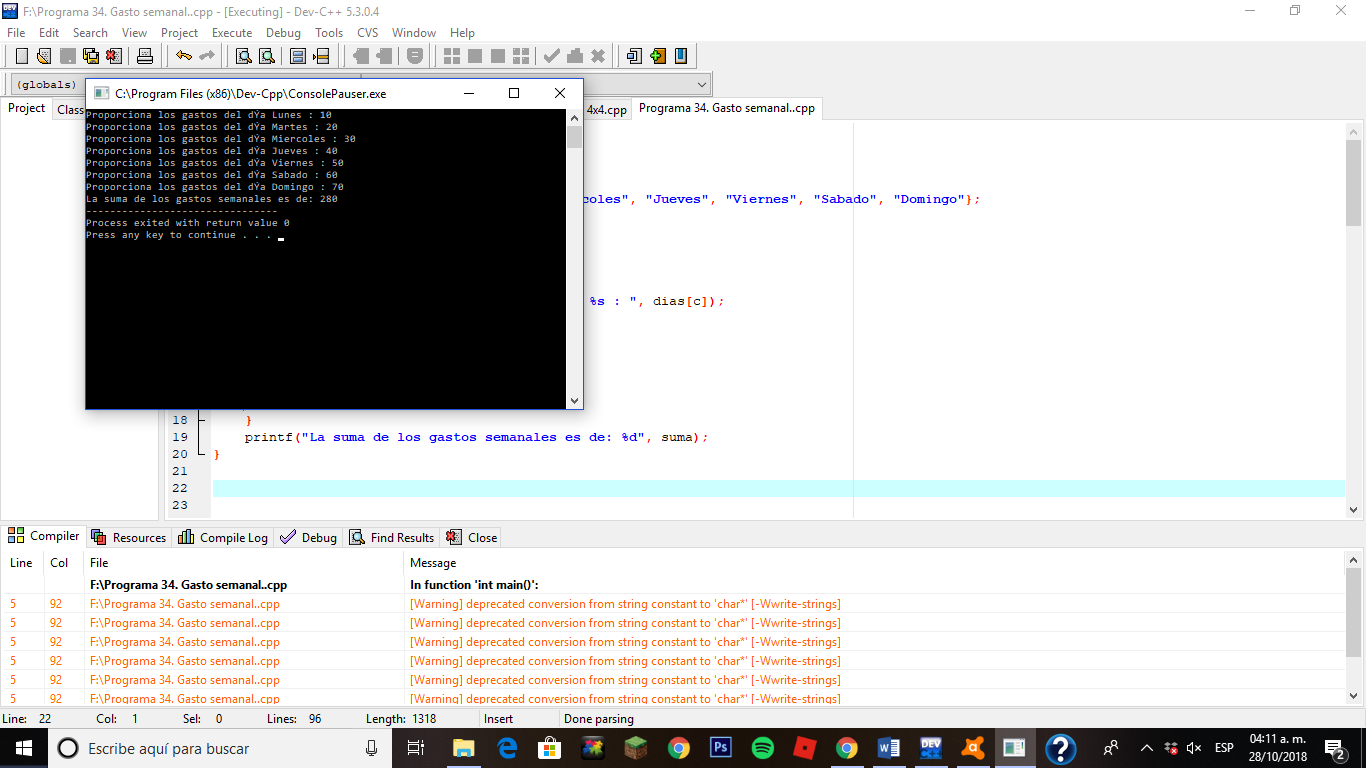
Durante esta práctica creamos tres nuevos programas con ayuda de las funciones vistas durante la clase. Los arreglos nos son de mucha ayuda al crear un programa queriendo optimizar la cantidad de memoria que nuestras variables ocupan en éste. A continuación se anexarán capturas donde se ejemplificarán los programas mostrados anteriormente.

1.- Promedio de cinco números con ayuda de arreglos.



2.- Suma de matrices 4x4. 

3.- Gastos semanales.

Conclusión.

Con la realización de esta práctica nos fue posible reforzar nuestros conocimientos previos sobre la programación con arreglos, tanto unidimensionales como bidimensionales, así como el manejo de apuntadores dentro de los programas. Además, con ayuda de lo visto en la práctica anterior (depuración) nos fue posible detectar rápidamente si nuestro programa contenía alguna falla y en qué parte se encontraba ésta.