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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | RODRIGUES ESPINO CLAUDIA |
| *Asignatura:* | FUNDAMENTOS DE PROGRAMACION |
| *Grupo:* | 4 |
| *No de Práctica(s):* | PRACTICA 12: Funciones |
| *Integrante(s):* | FLORES MENDOZA OLGA |
|  |  |
| *No. de Equipo de cómputo empleado* | 44 |
| *Semestre:* | 2019-2 |
| *Fecha de entrega:* | 12 DE MAYO DE 2019 |
| *Obervaciones:* |  |
|  |  |

ARREGLOS

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**PRACTICA 12**

**OBJETIVOS:**

Elaborar programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

**ACTIVIDADES:**

* Implementar en un programa en C la solución de un problema dividido en funciones.
* Elaborar un programa en C que maneje argumentos en la función principal.
* En un programa en C, manejar variables y funciones estáticas.

**INTRODUCCION:**

Un programa en lenguaje C consiste en una o más funciones. En lenguaje C la función principal se llama main. Cuando se ordena la ejecución del programa, se inicia con la ejecución de las instrucciones que se encuentran dentro de la función main, y ésta puede llamar a ejecutar otras funciones, que a su vez éstas pueden llamar a ejecutar a otras funciones, y así sucesivamente.

La sintaxis básica para definir una función es la siguiente:

**valorRetorno nombre (parámetros);**

**{ // bloque de código de la función }**

Una función está compuesta por tres elementos: el nombre de la función, los parámetros que recibe la función y el valor de retorno de la función; finaliza con (;).

El nombre de la función se refiere al identificador con el cual se ejecutará la función; se debe seguir la notación de camello. Una función puede recibir parámetros de entrada, los cuales son datos de entrada con los que trabajará la función e indicando su tipo de dato, de la siguiente forma:

**(tipoDato nom1, tipoDato nom2, tipoDato nom3…)**

El tipo de dato puede ser cualquiera de los vistos hasta el momento y el nombre debe seguir la notación de camello. El valor de retorno de una función indica el tipo de dato que va a regresar la función al terminar el bloque de código de la misma.

Las funciones definidas en el programa no necesariamente deberán ser declaradas; esto dependerá de su ubicación en el código.

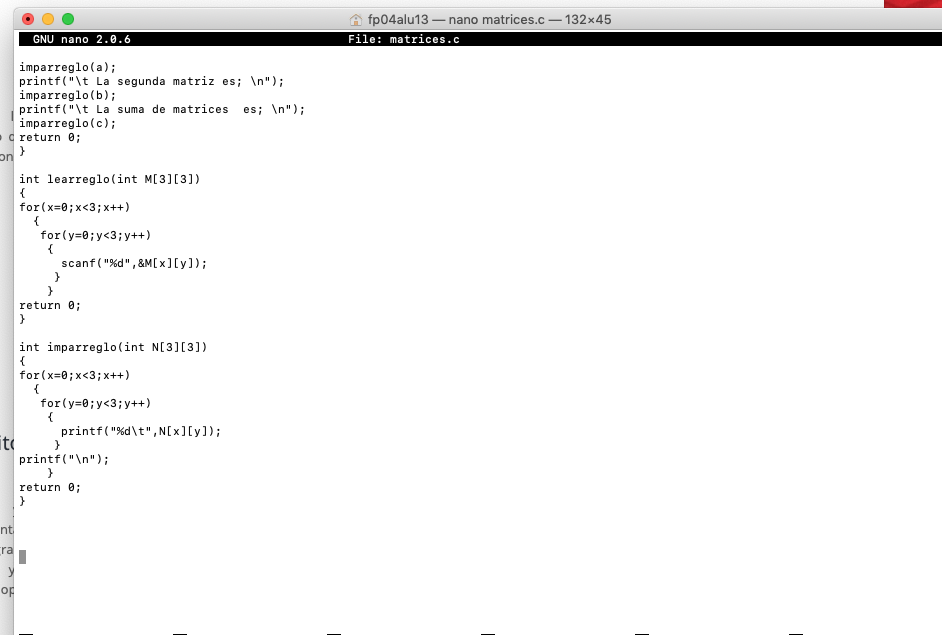
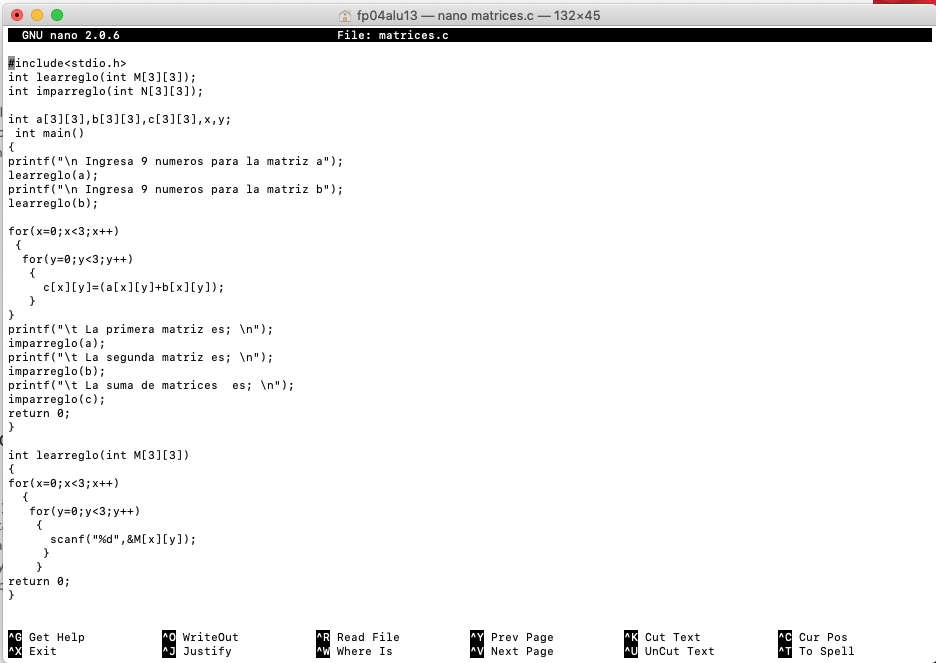
Ámbito o alcance de las variables

Las variables declaradas dentro de un programa tienen un tiempo de vida que depende de la posición donde se declaren. En C existen dos tipos de variables con base en el lugar donde se declaren: variables locales y variables globales. Las variables que se declaren dentro de cada función se conocen como variables locales (a cada función). Estas variables existen al momento de que la función es llamada y desaparecen cuando la función llega a su fin. Las variables que se declaran fuera de cualquier función se llaman variables globales. Las variables globales existen durante la ejecución de todo el programa y pueden ser utilizadas por cualquier función.

**DESARROLLO DE ACTIVIDADES:** Ser desarrollaron tres programas haciendo uso de las funciones. Uno fue creado en la terminal usando GNU y los dos restantes en Dev-c++.

En el programa 1 y 2 se declararon los funciones con **int,** mientras que el tercero fue declara la función con **long**.

**PROGRAMA 1:** Este programa está editado en la terminal y realiza la tara de pedir al usuario una matriz w para sumarla con una matriz k y dar el resultado en una tercera matriz (q). Se emplearon 2 funciones una que leía los datos y otra que imprimía los datos, la estructura for y 3 arreglos.



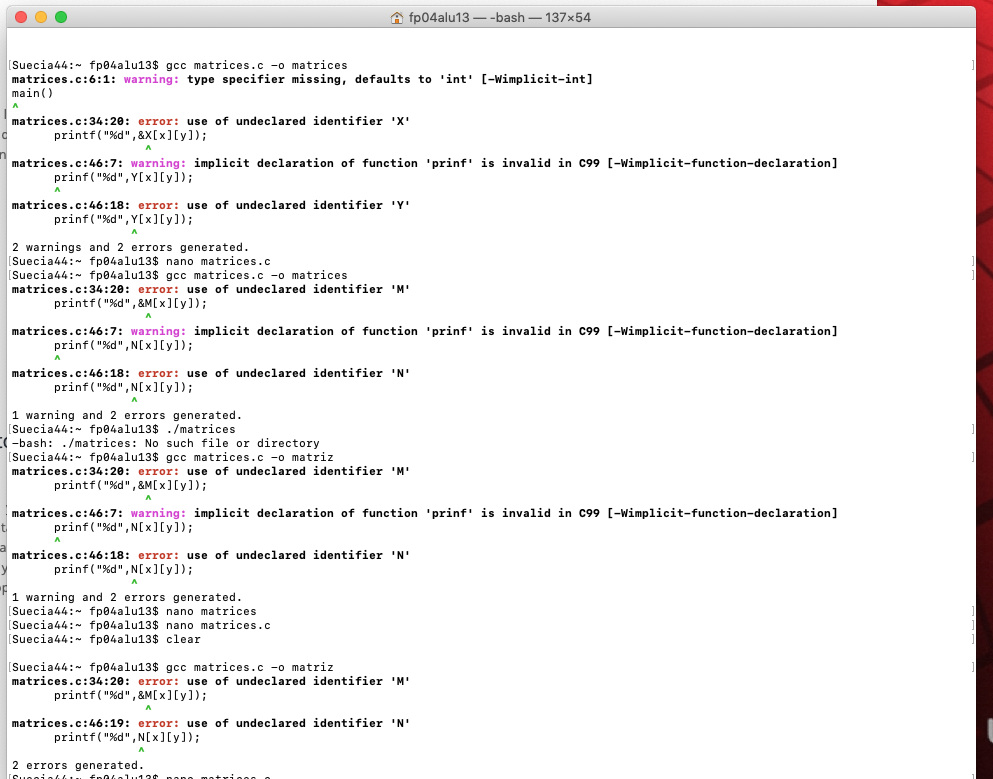
Función 1

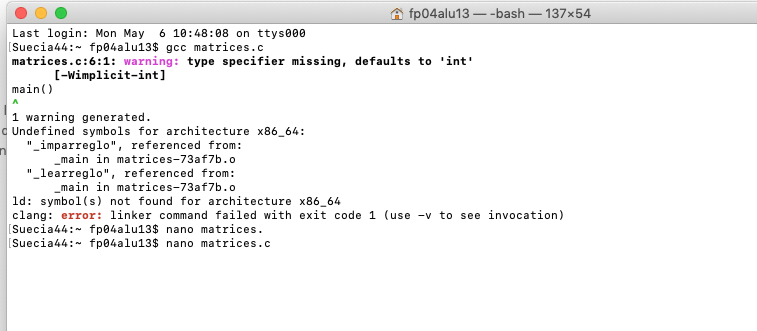
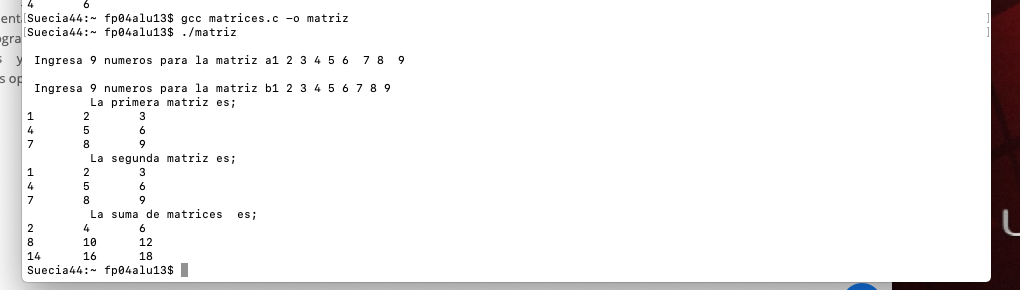
Función 2

ARREGLOS

FUNCIONES

Se presentaron varios errores con el programa pero giraban en torno de declarar la función adecuadamente y escribirla bien en el resto del programa. Los errores fueron los siguientes.

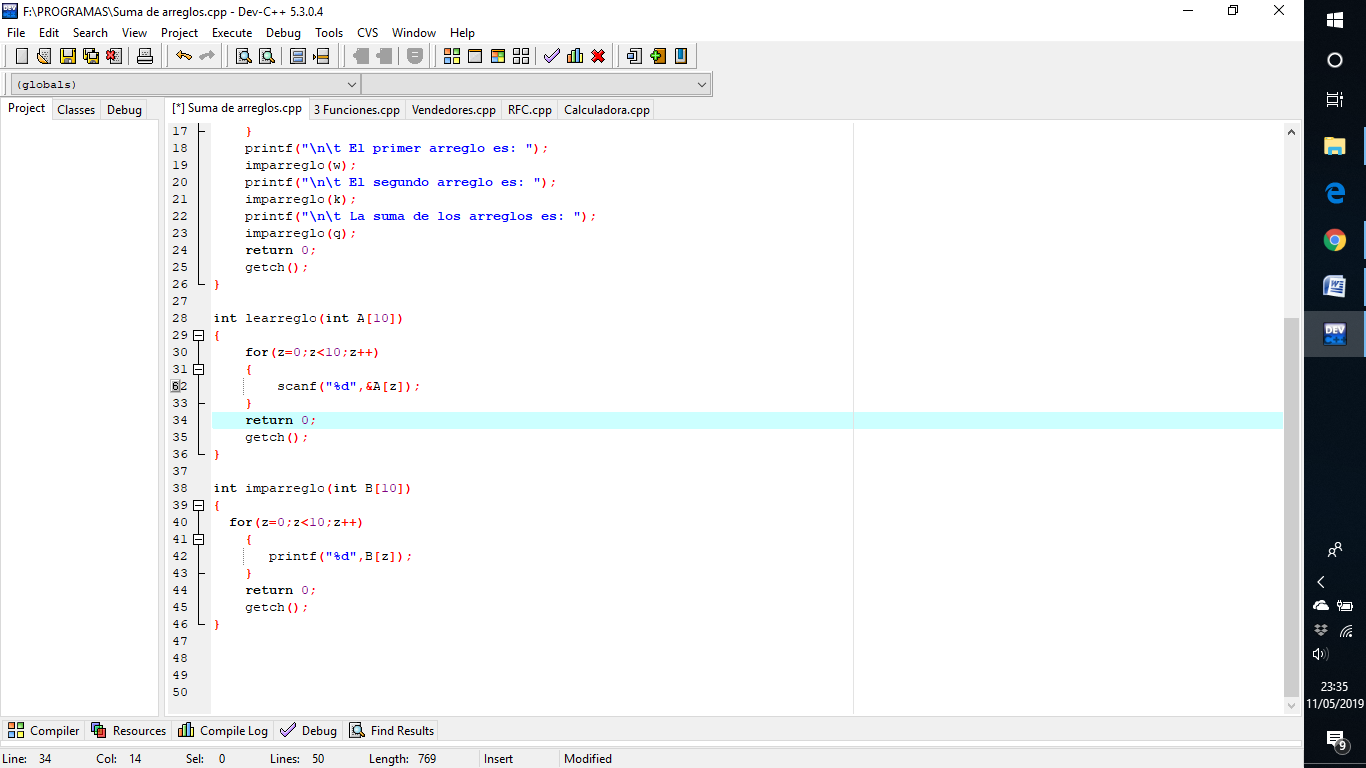
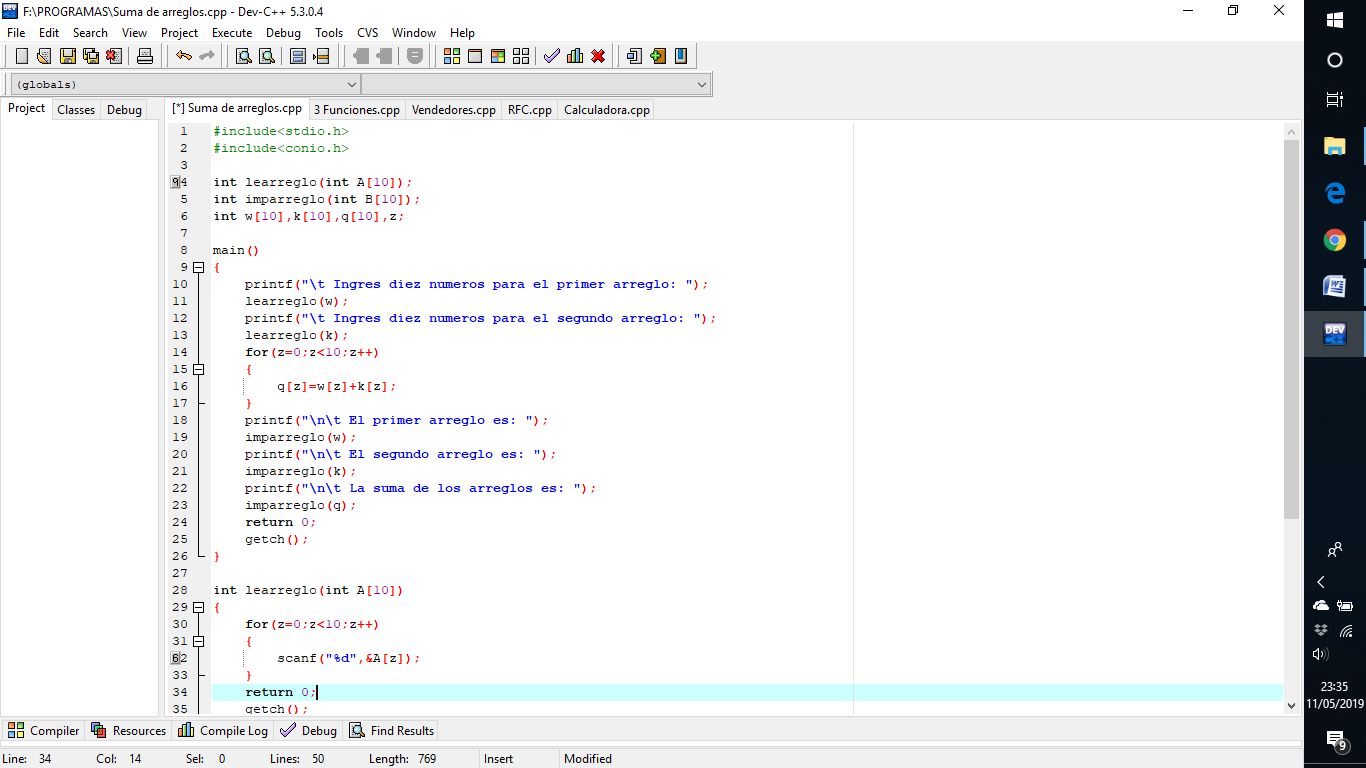




Se compilo el programa una vez más, al no presentar errores los siguiente fue la ejecución del programa para verificar si realizaba adecuadamente su tarea. El resultado se muestra a continuación.

**PROGRAMA 2:** Este programa pide al usuario 10 número para un arreglo y diez para otro, procediendo a realizar la suma de estos e imprimirla. Se hizo uso de 2 funciones y 3 arreglos, así como de estructuras de repetición for. Este programa fue editado en Dev-c++

Las funciones realizan la misma tarea que las que aparecen en el programa anterior.

****

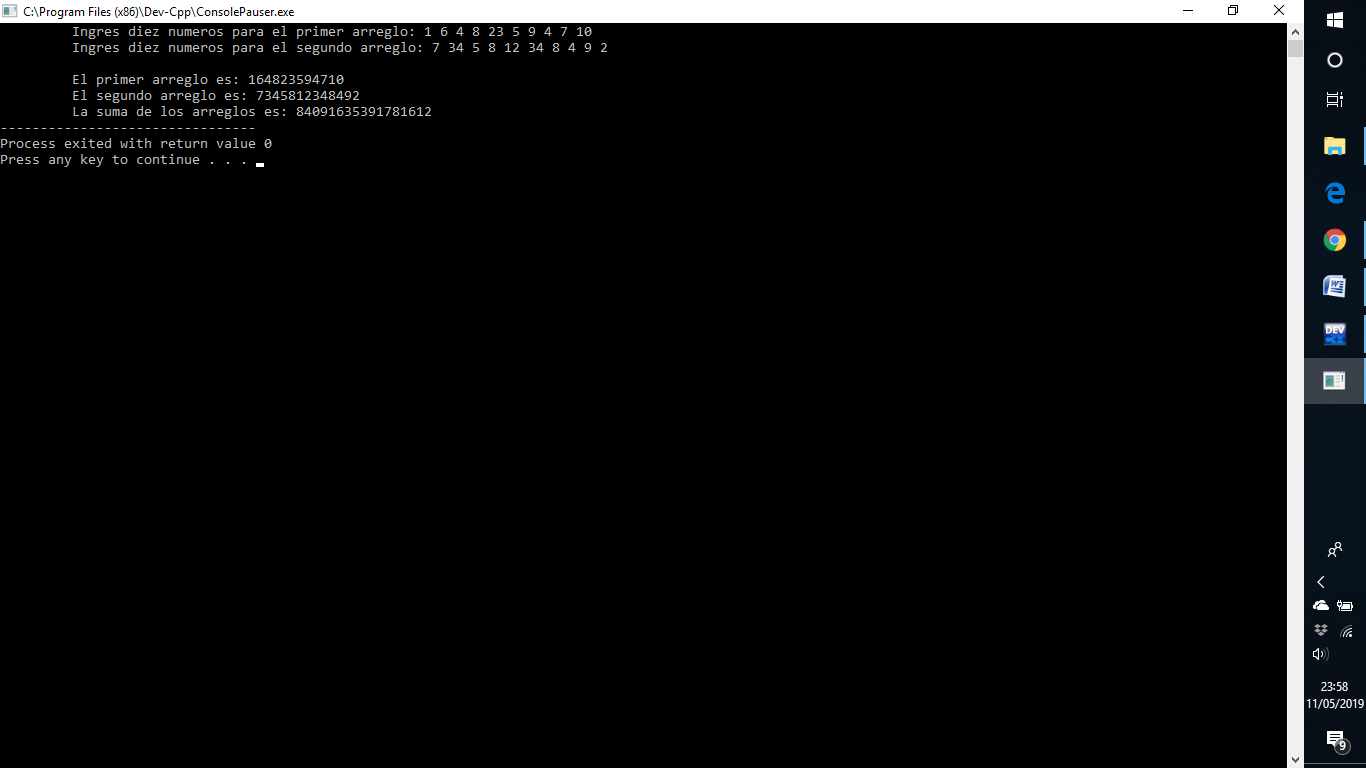
Función 2

Función 1

FUNCIONES

ARREGLOS

Se realiza la compilación, se corrigieron los errores presentes ; el siguiente paso fue la ejecución del programa.

****

PROGRAMA 3: Este último programa es una calculadora, le da la opción al usuario de escoger la operación aritmética que quiera y procede a realizarla después de ser escogida con datos proporcionados por el usuario. Se utilizo 4 funciones, switch-case, long como valor, , estructuras de selección, entre otras.

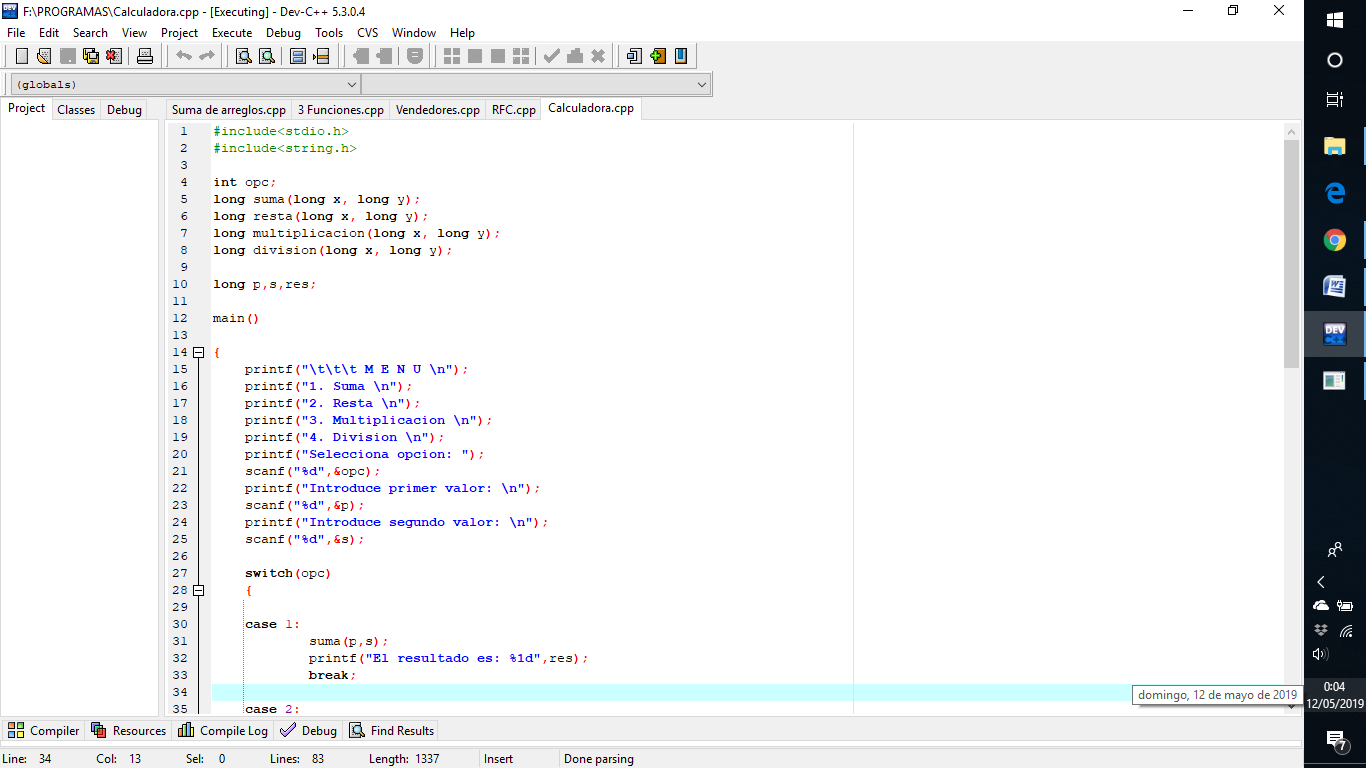
Cada una de las funciones realizaba una tarea distinta, por ejemplo:

long suma(long x, long y); = Realiza una suma de los datos

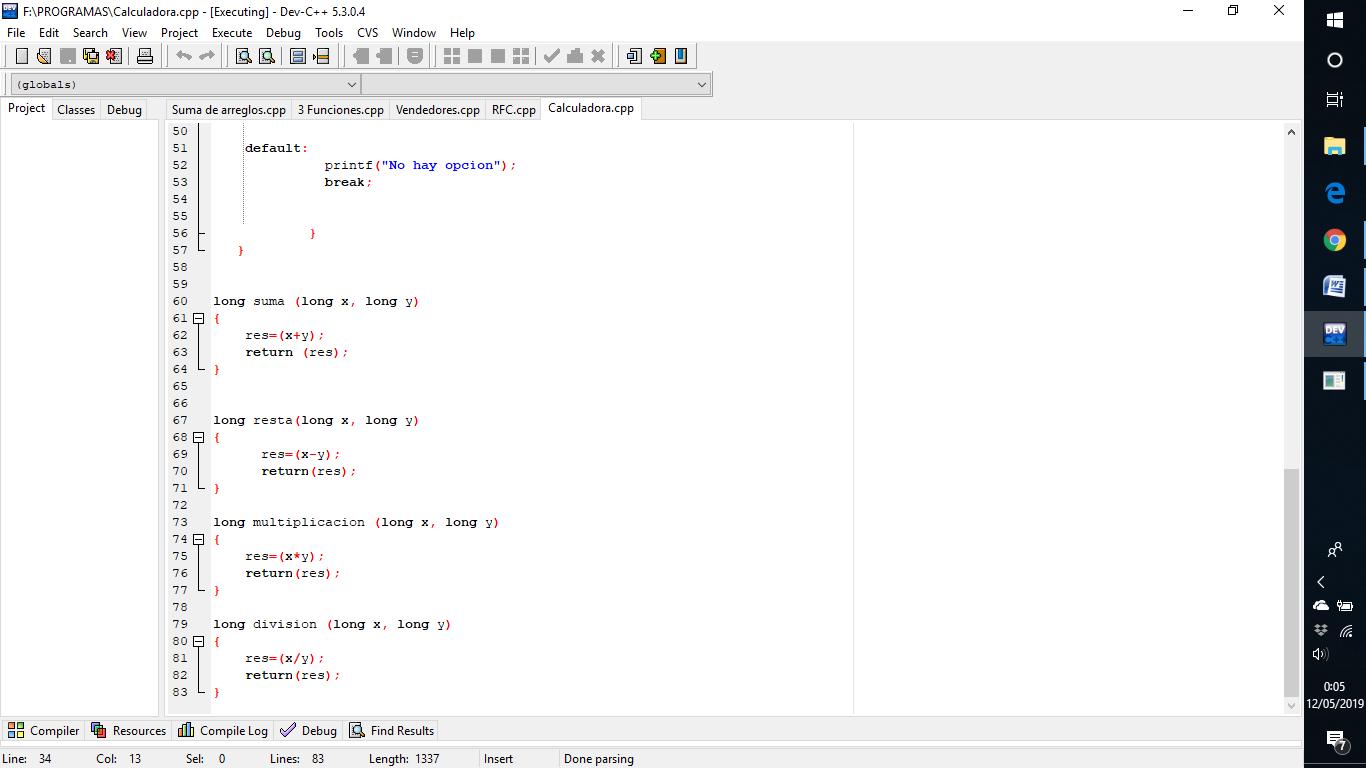
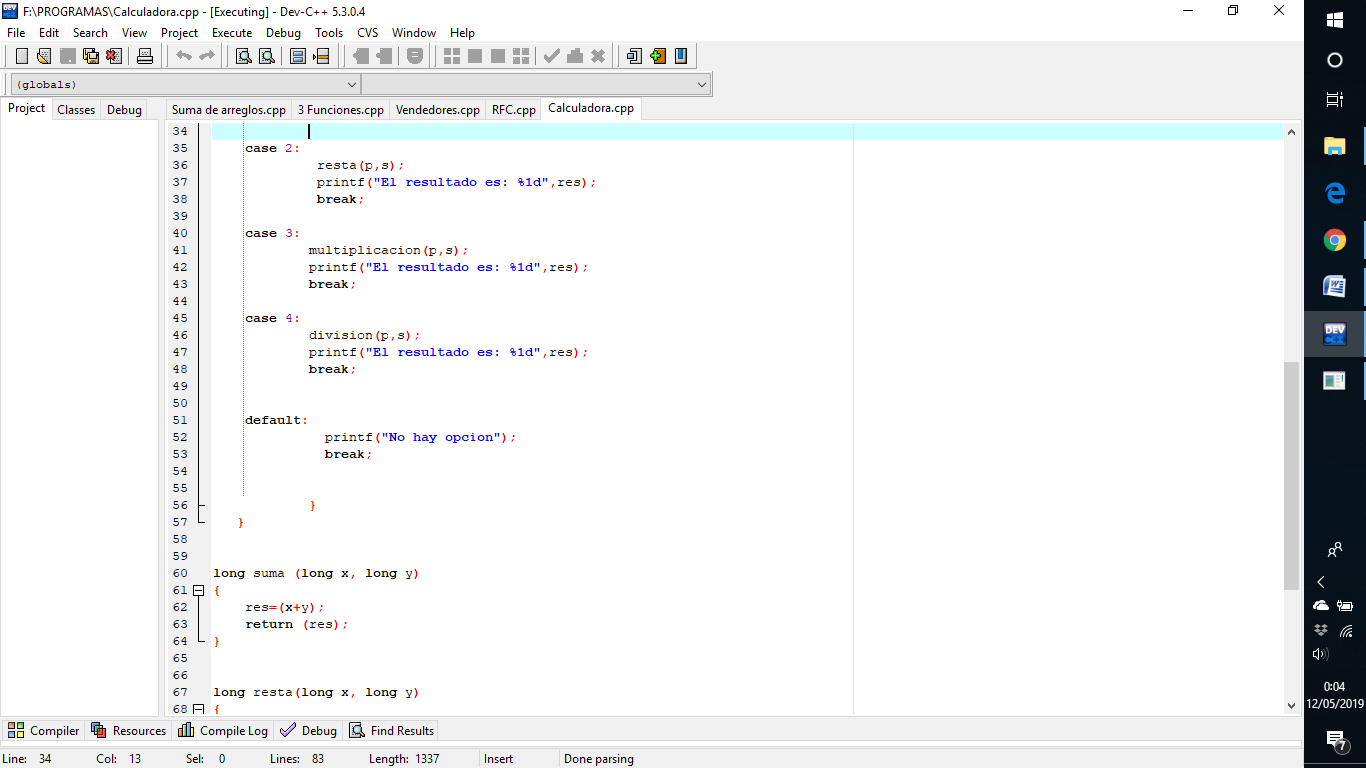
long resta(long x, long y); = Realiza una resta con los datos proporcionados

long multiplicacion(long x, long y); = Realiza una multiplicación de los números

long division(long x, long y); = Realiza una división de los números dados.

****

FUNCIONES

****

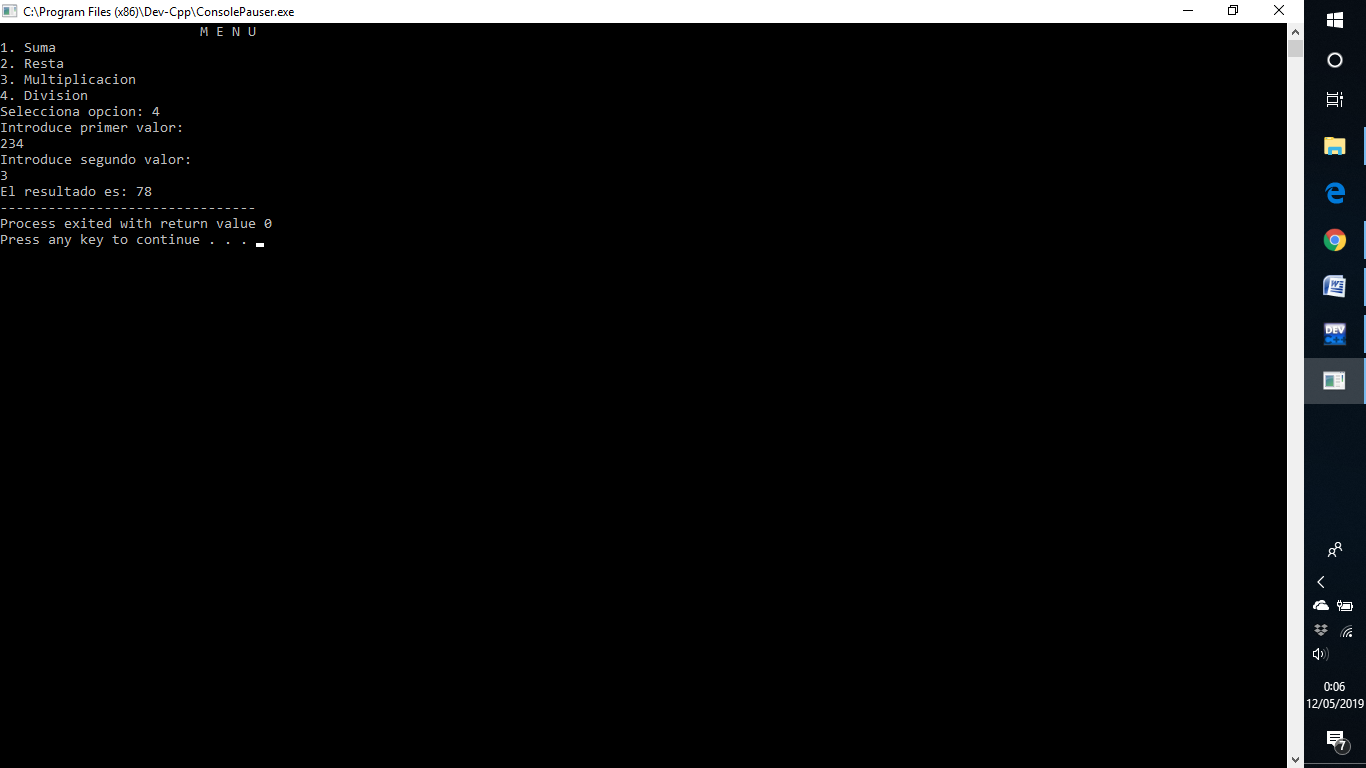
Función 4

Función 3

Función 2

Función 1

La creación de la calculadora no fue difícil pero si elaborada, se compilo y de ejecuto, comprobando la veracidad de su programación. A continuación se muestra la ejecución del programa.

****

**CONCLUCIONES:**

En un principio no comprendía la tarea de las funciones, su declaración en el programa era confusa con los arreglos, pero después de la práctica y la realización de programas comprendí su sintaxis y su uso. La función realiza la tarea, operación, etc. que el programador le asigne es por esto que se debe de saber que se quiere realiza y como se hará, también se debe considerar los parámetros, de estos dependerá que tanta libertad se le da al usuario para ingresar un dato. Es importante cerrar la función principal ( main() ) antes de abrir y una nueva función, de lo contrario el programa no se podrá ejecutar. Es muy importante abrir y cerrar cada función. Estas son de gran ayuda para programas más complejos y con tareas mayores.