

**Objetivo:**

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.

**Introducción**

**Funciones**

La sintaxis básica para definir una función es la siguiente:

valorRetorno nombre (*parámetros*){

// bloque de código de la función

}

El nombre de la función se refiere al identificador con el cual se ejecutará la función; se debe seguir la notación de camello.

Valor de Retorno: valor que el programa regresa

tipo\_dato nombre\_funciòn(int valor, int valorDos)

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.

strlen es una función que recibe como parámetro un arreglo de caracteres y regresa como valor de retorno un entero que indica la longitud de la cadena. La función se encuentra dentro de la biblioteca string.h, por eso se incluye ésta al principio del programa.

**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.

*strlen*es una función que recibe como parámetro un arreglo de caracteres y regresa como valor de retorno un entero que indica la longitud de la cadena. La función se encuentra dentro de la biblioteca *string.h*, por eso se incluye ésta al principio del programa.

**Argumentos para la función main**

Como se mencionó́ anteriormente, la firma de 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.

La función main también puede recibir parámetros. Debido a que la función main es la primera que se ejecuta en un programa, los parámetros de la función hay que enviarlos al ejecutar el programa. La firma completa de la función main es:

int main (int argc, char \*\* argv);

La función main puede recibir como parámetro de entrada un arreglo de cadenas al ejecutar el programa. La longitud del arreglo se guarda en el primer parámetro (argument counter) y el arreglo de cadenas se guarda en el segundo parámetro (argument vector). Para enviar parámetros, el programa se debe ejecutar de la siguiente manera:

* En plataforma Linux/Unix ./nombrePrograma arg1 arg2 arg3 ...
* En plataforma Windows nombrePrograma.exe arg1 arg2 arg3 ...

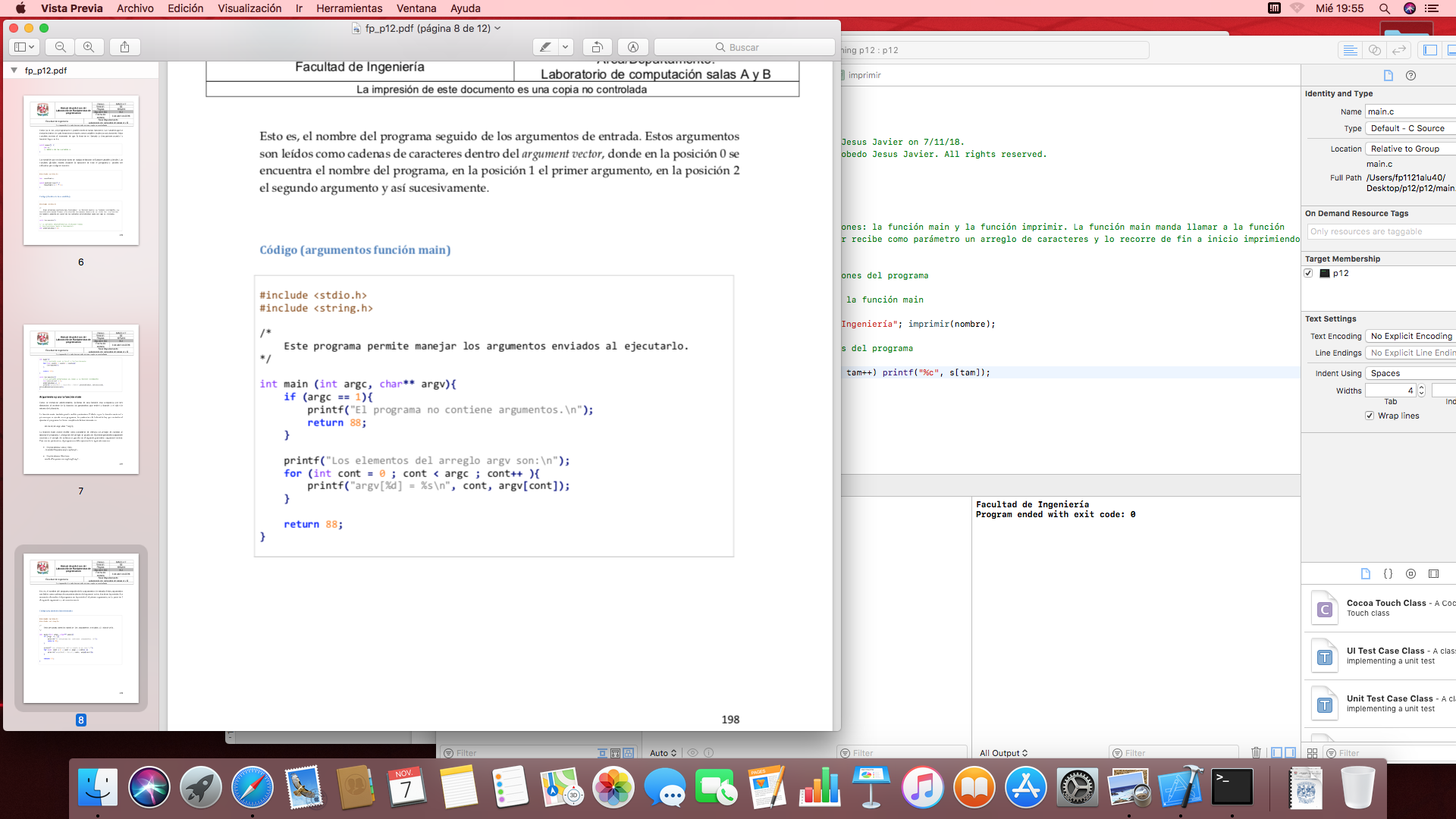
**Funciones estáticas**

La definición de una función puede tener el prefijo “static”. Cuando una función se declara como estática, tan sólo puede ser invocada desde el fichero en el que está definida. Este mecanismo, por tanto, puede interpretarse como una forma primitiva de restringir el acceso a una función, pero dista mucho del mecanismo de tres niveles (público, privado y protegido) presente en Java.

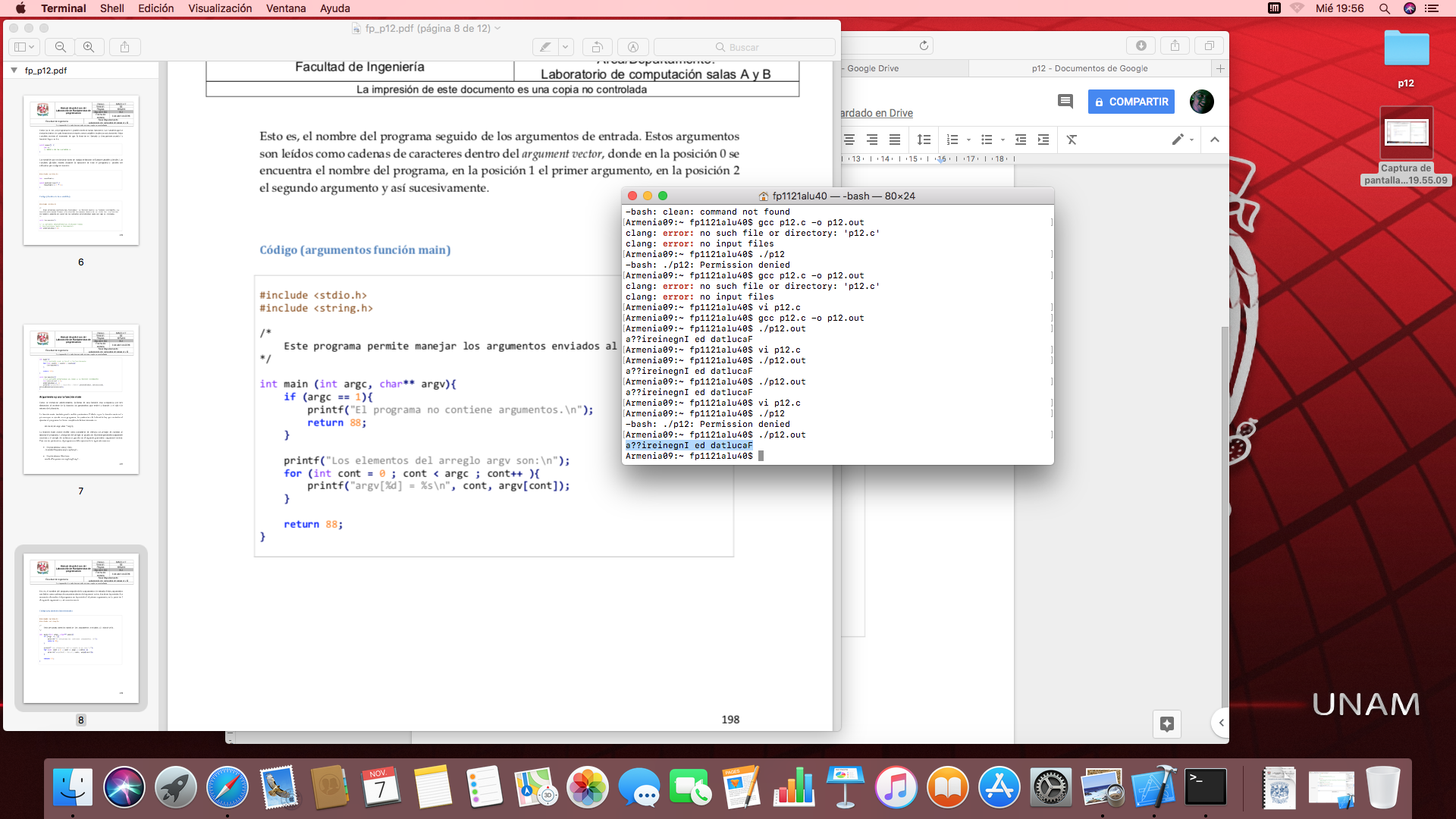
Cuando se desarrollan aplicaciones de gran tamaño, se suelen establecer políticas para el uso de prototipos. Por ejemplo, para poder invocar cualquier función en cualquier parte del código de un fichero, se suelen colocar al comienzo del mismo los prototipos de todas las funciones que contiene. El siguiente ejemplo muestra un ejemplo de esta política:

**Ejercicios**

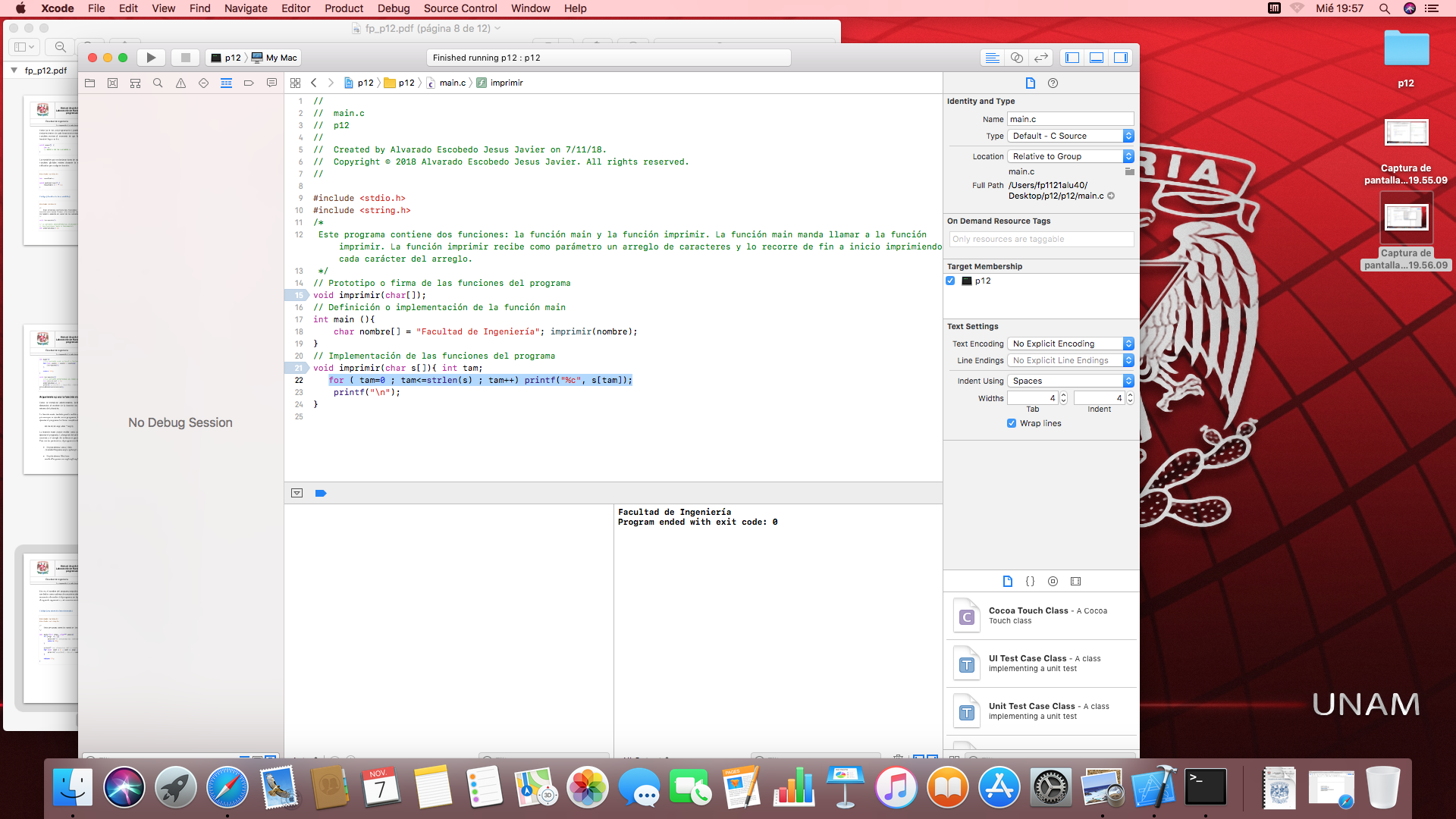
con este código



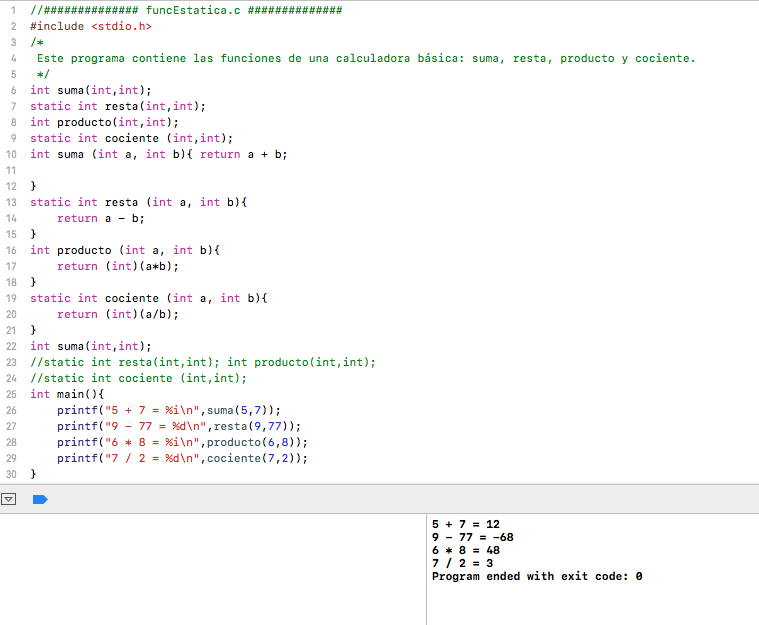
imprime lo siguiente

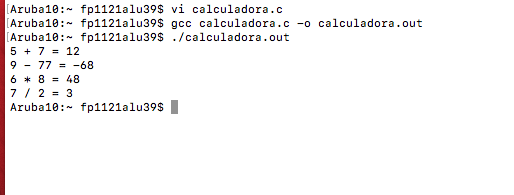


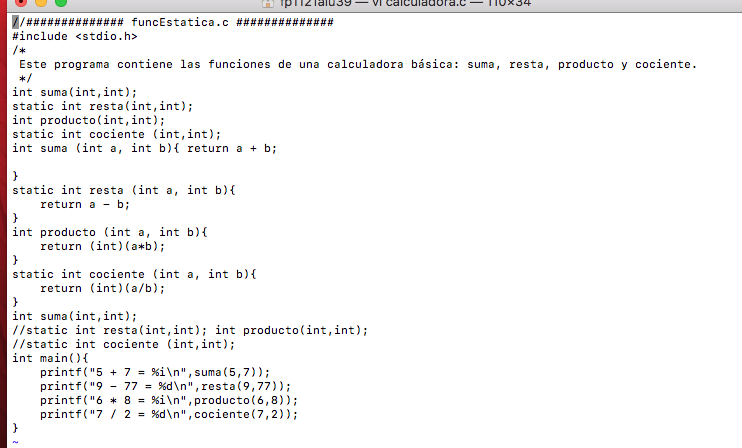
arreglando el ciclo for, aparece correcto la cadena de caracateres



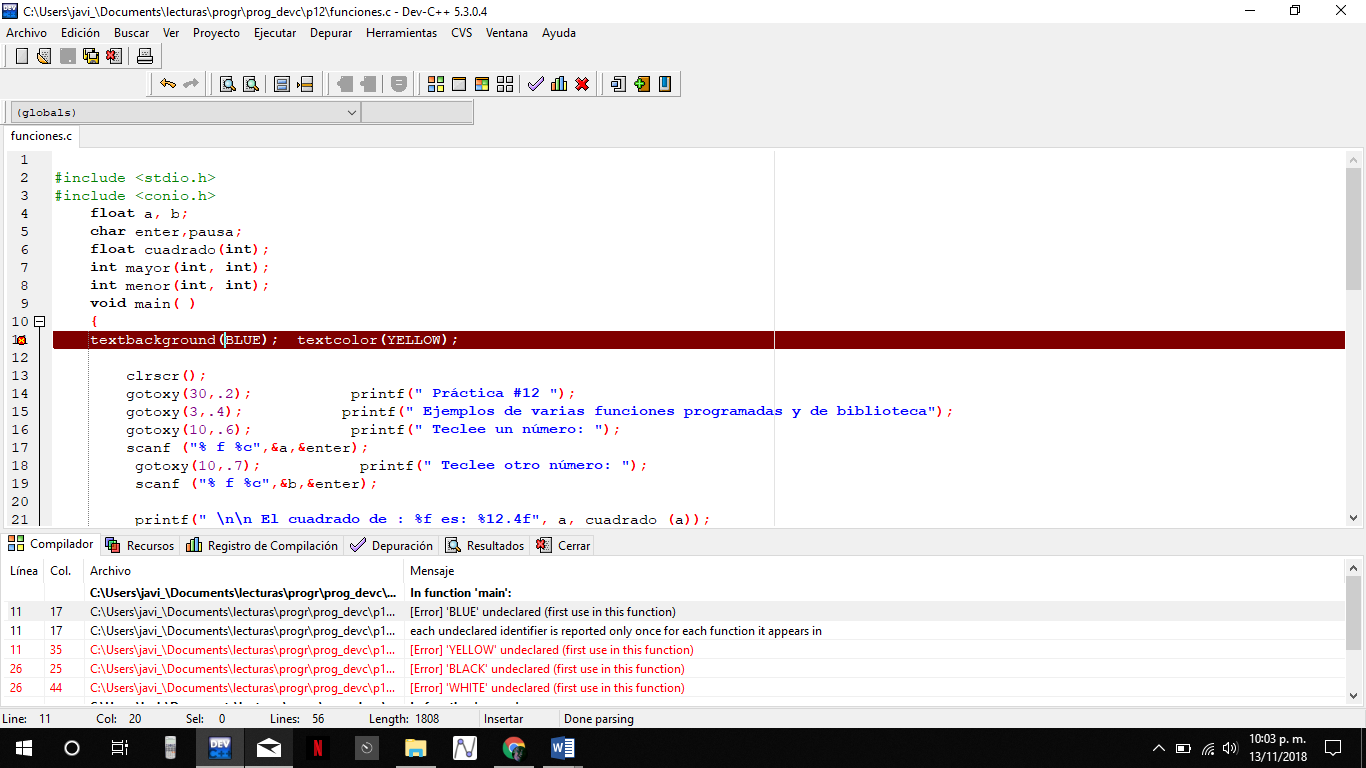
En otro programa.

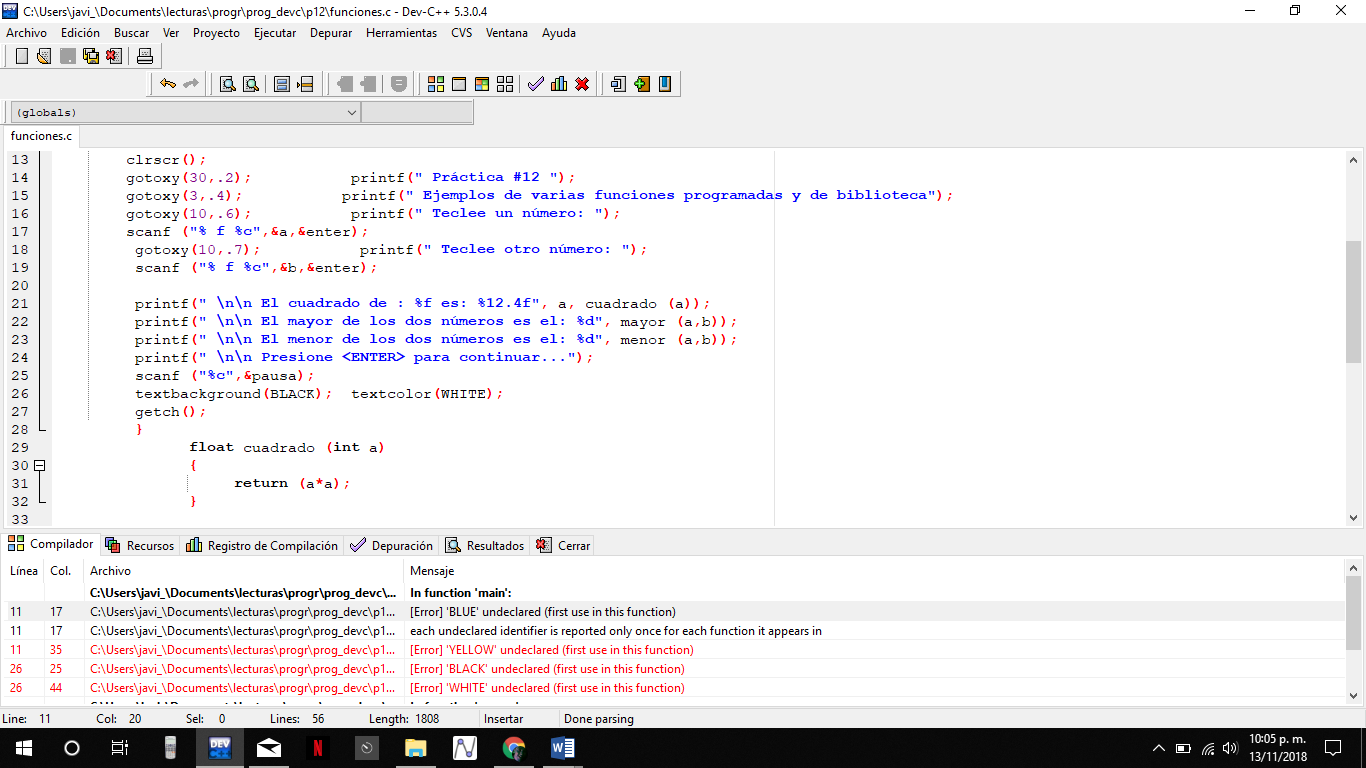


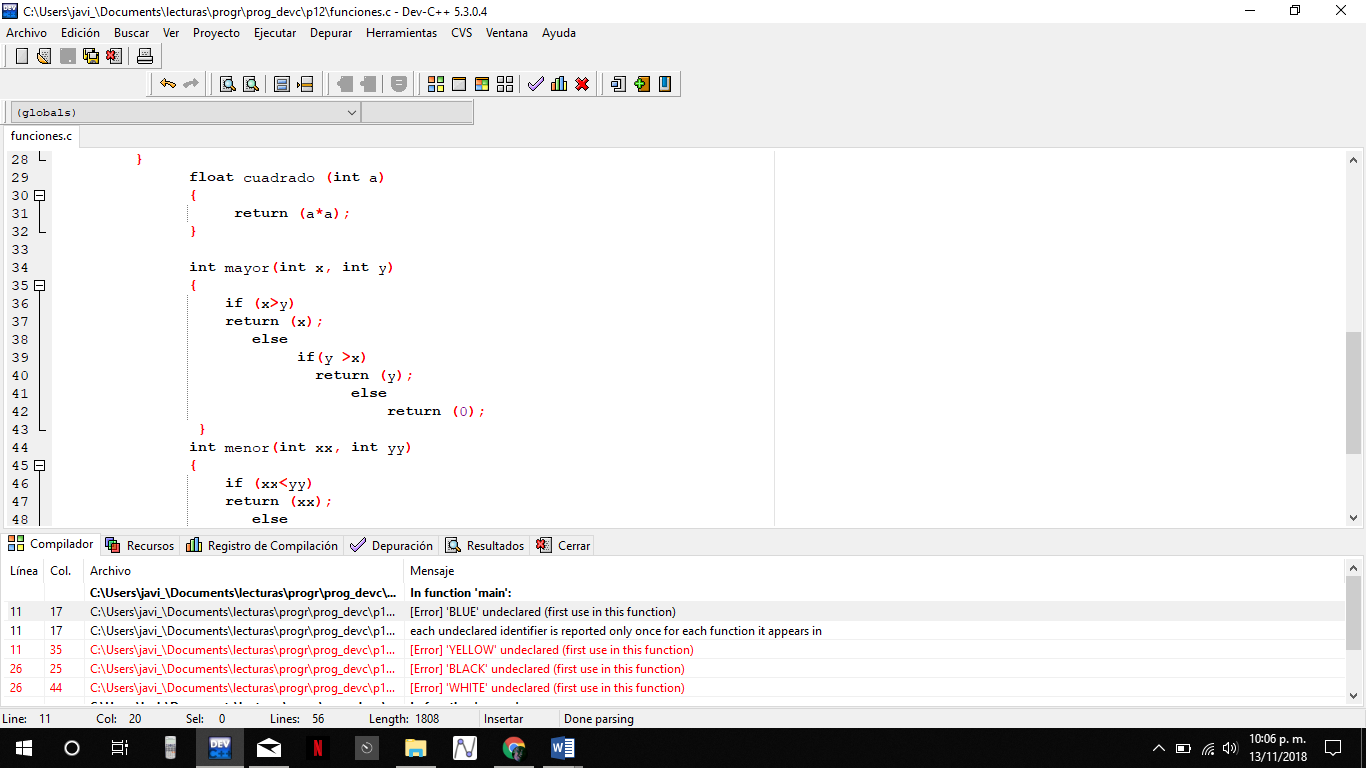




1.- Realice el programa de **funciones que decida crear usted asigna el nombre**









Debe mejorarlo, modificarlo o crear su propia solución

Describa:

a) los errores que obtuvo (en caso de haberlos)

b) el desplegado obtenido

Marque en el programa donde se realizan:

a) las llamadas a las funciones

b) la declaración de las funciones

c) la definición de las funciones

Genere el programa como un proyecto y separe las funciones, una por archivo, cambie el tipo de datos en la declaración de funciones a estáticas y describa qué pasa.

# **Bibliografías.**

http://profesores.fi-b.unam.mx/cintia/c\_basico.pdf

http://michelletorres.mx/arreglos-unidimensionales-en-c/

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.

https://www.aprenderaprogramar.com/index.php?option=com\_content&view=article&id=901:arrays-unidimensionales-arreglos-o-formaciones-en-c-declaracion-y-limites-indices-ejemplos-cu00512f&catid=82&Itemid=210