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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Garcia Morales Karina |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | 22 |
| *No de Práctica(s):* | Practica 12 |
| *Integrante(s):* | Ramos Orozco Daniel |
|  |  |
|  |  |
| *Semestre:* | 2018-2 |
| *Fecha de entrega:* | 15-05-2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**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.

**Funciones:**

Lo primero que vimos y en este caso que recordamos es que solo puede haber un main en las funciones y ese main es en la función principal.

**Strlen:** recibe como parámetro un arreglo de caracteres y regresa como valor un entero que indica la longitud de la cadena. La función se encuentra dentro de la biblioteca string.h.

**Variables globales:**

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.

**Variables locales:**

Las variables que se declaren dentro de cada función se conocen como variables locales. Estas variables existen al momento de que la función es llamada y desaparecen cuando la función llega a su fin.

La sintaxis básica de función es:

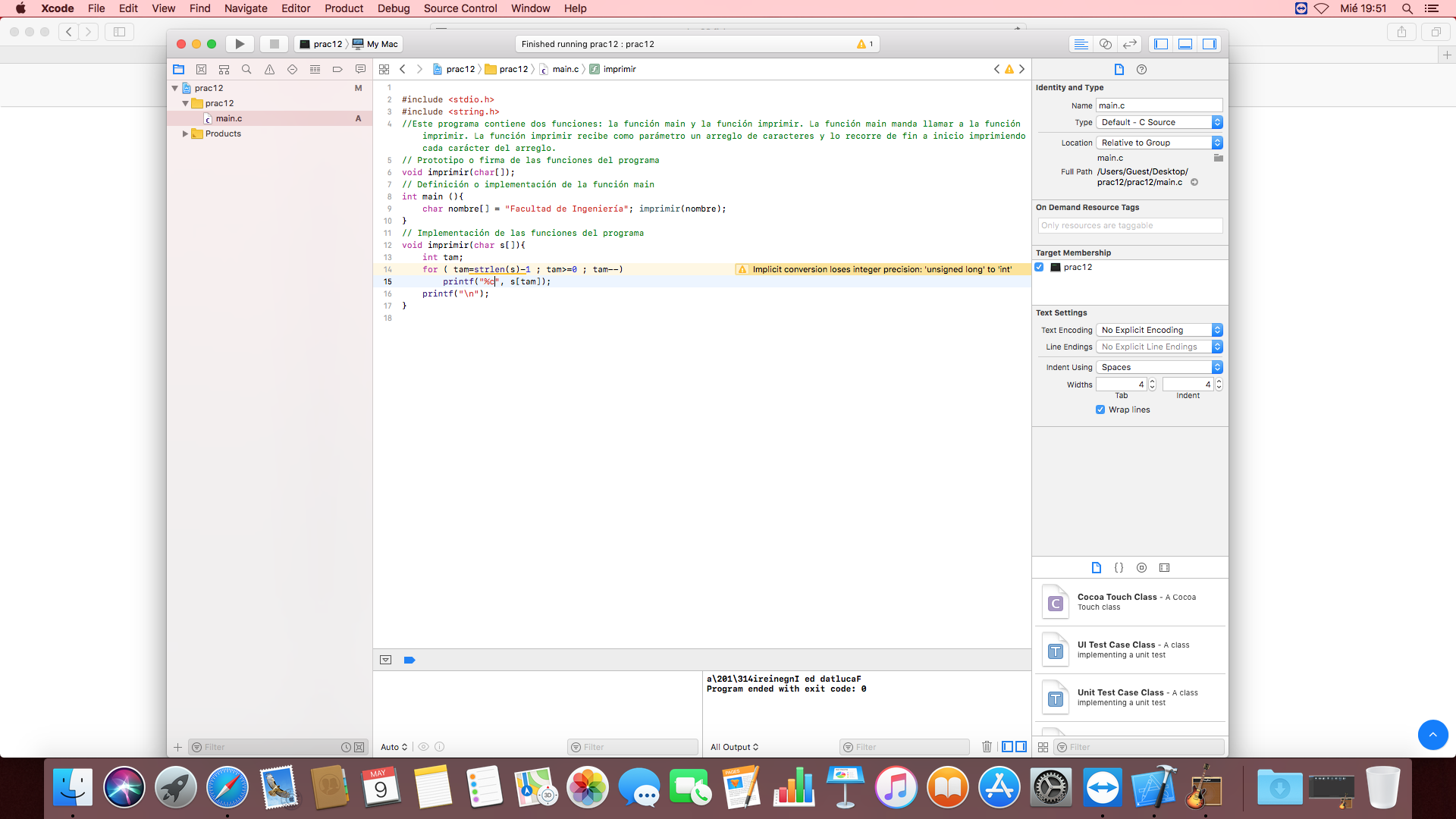
valorRetorno nombre (parámetros){

// bloque de código de la función

}

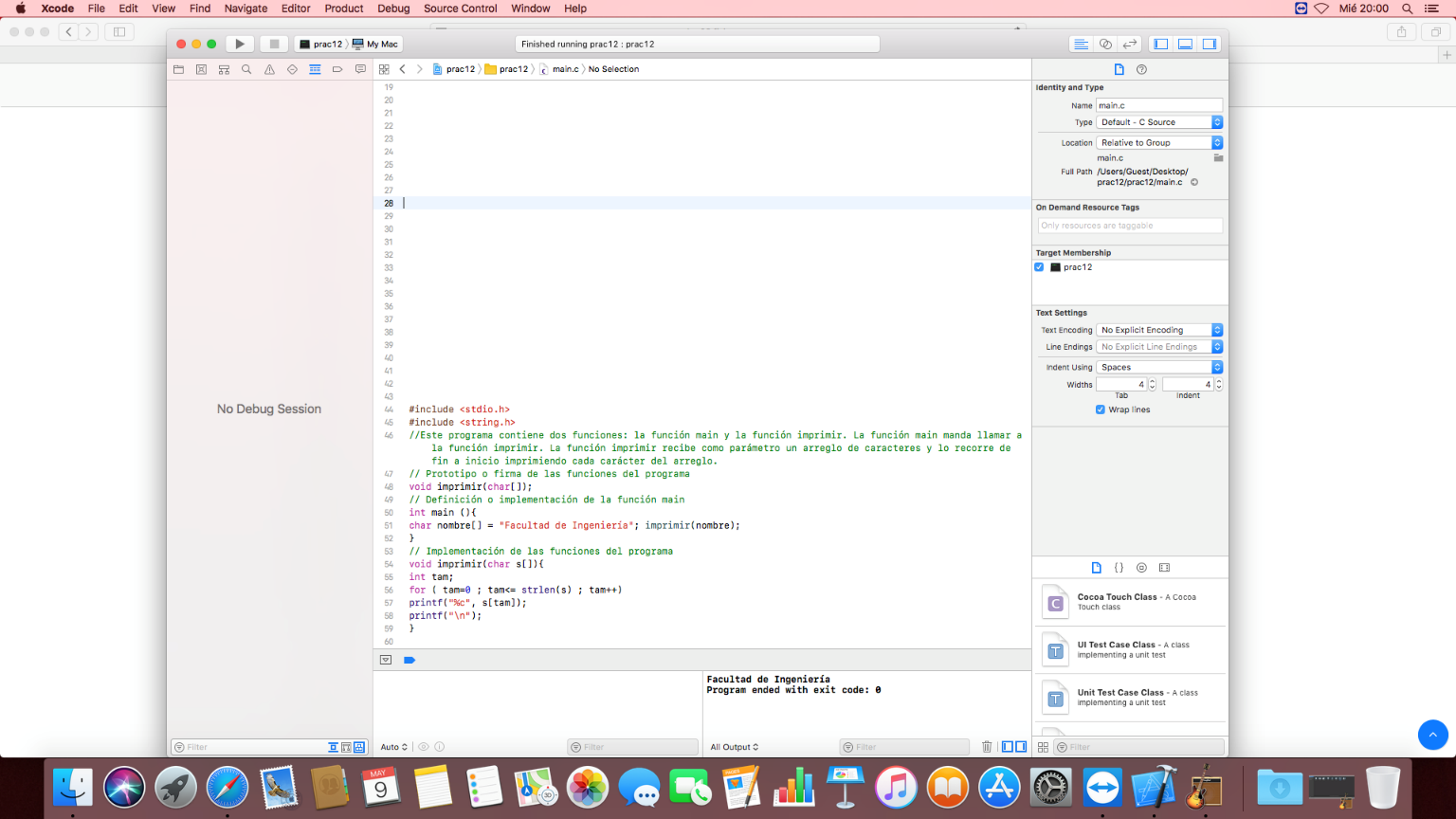
Aunque varía dependiendo de lo que se necesite, ya que en algunos casos no es necesario poner los parámetros de la función.

Primero vimos el siguiente ejemplo:

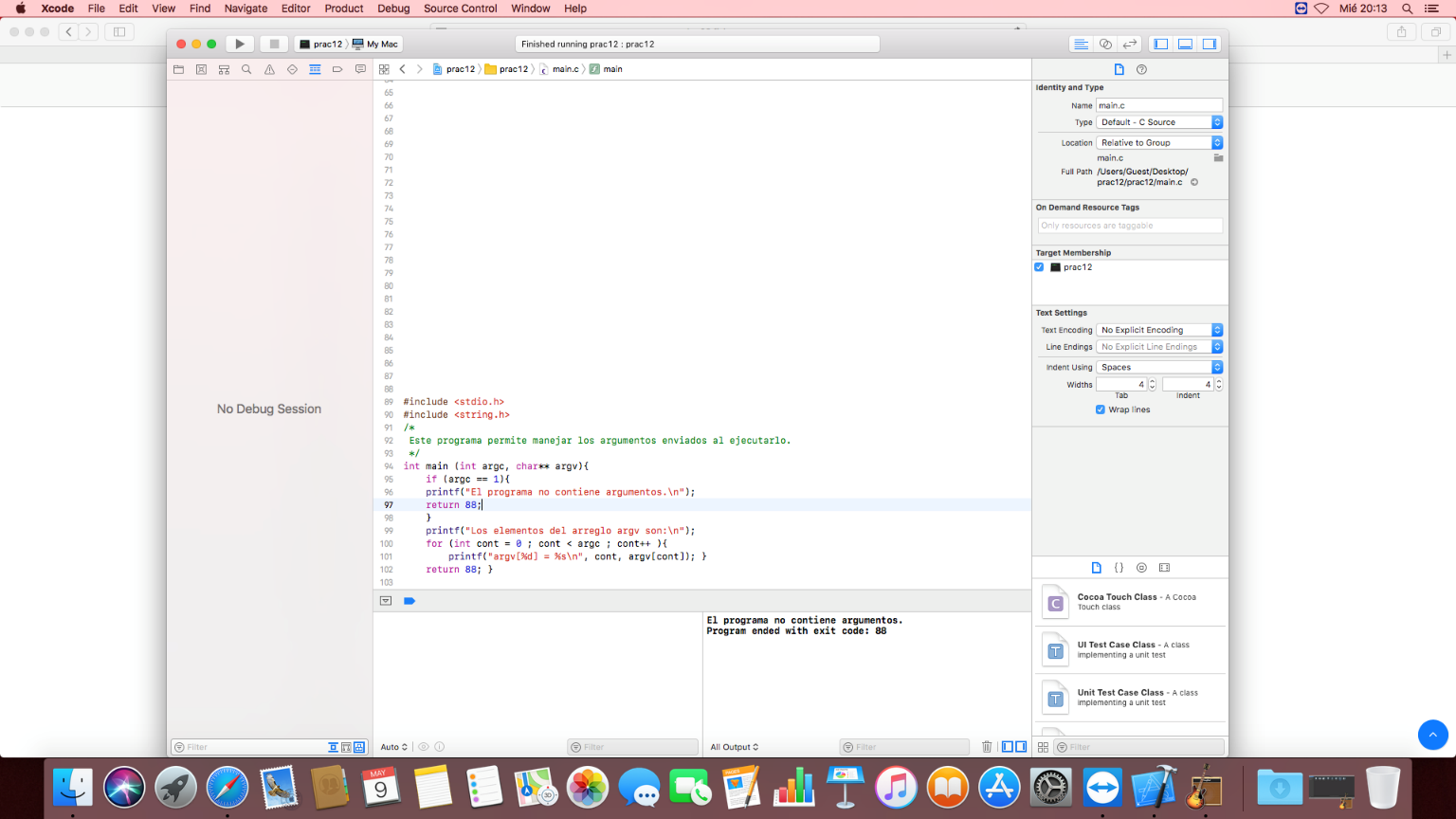


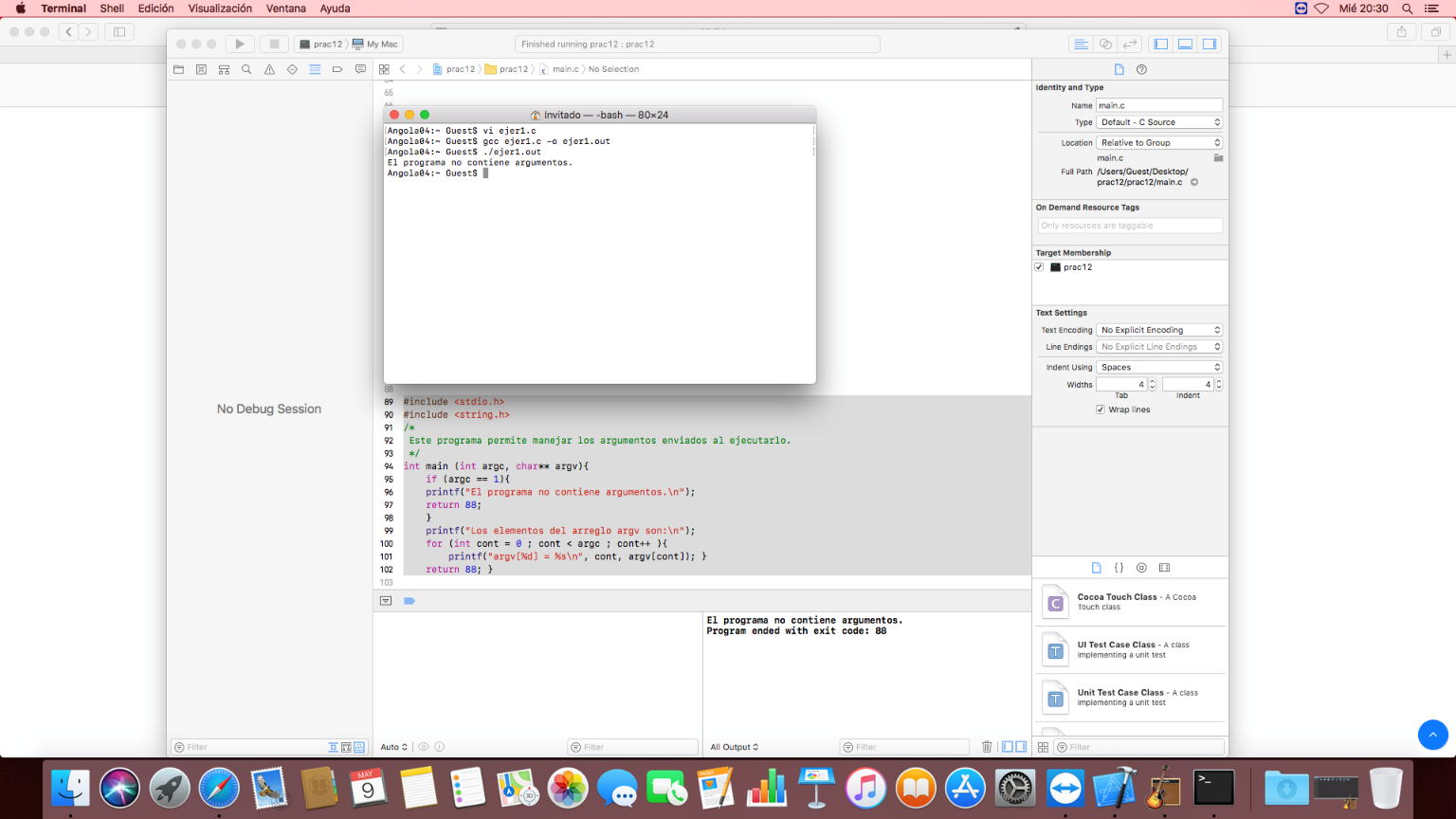
En el cual a base de funciones, arreglos y ciclos, lee una cadena, que lo que hace es que te de la cadena pero al revés.

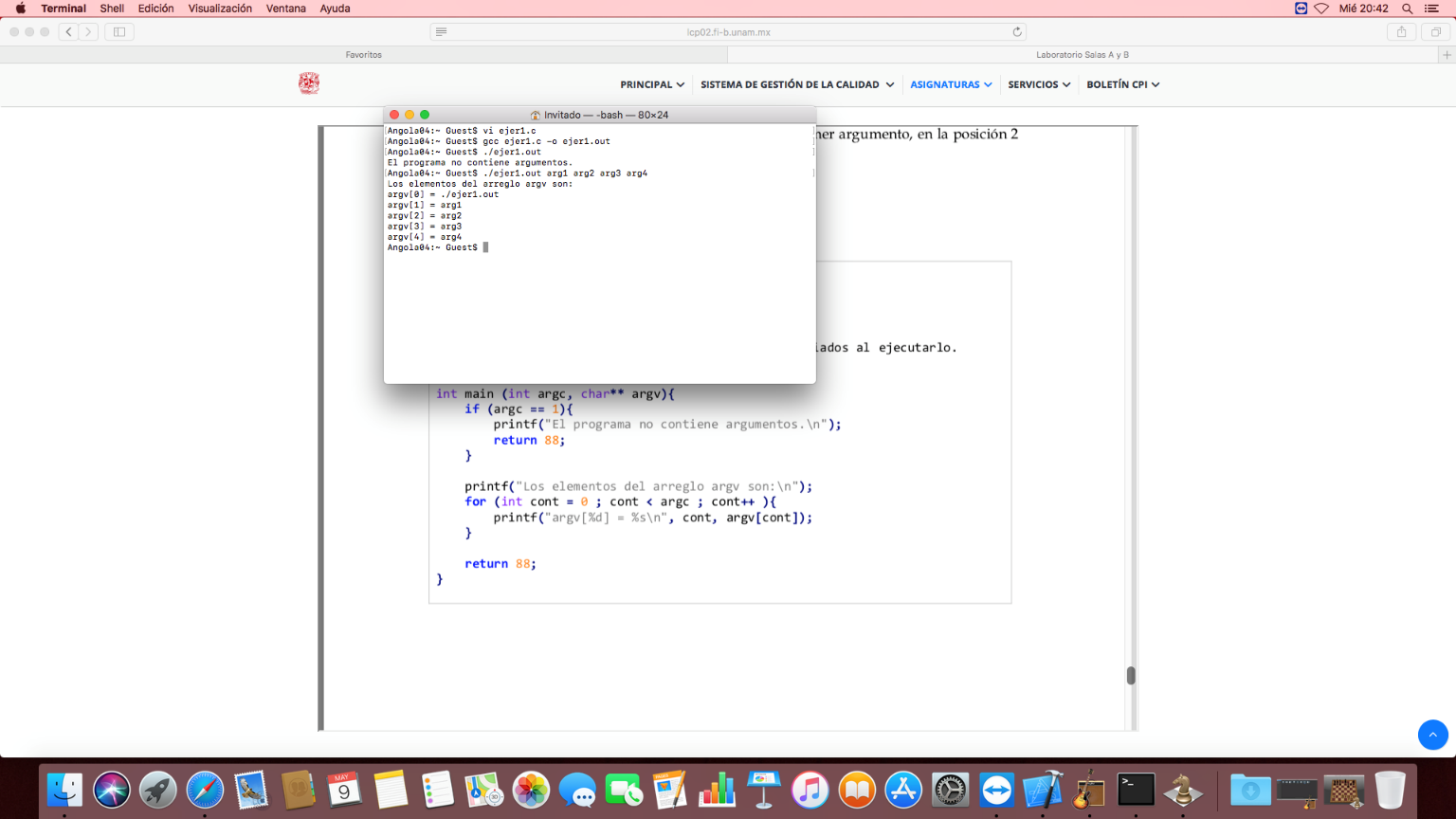
Despues vimos un ejemplo de funciones utilizando la funcion strlen haciendo uso de lo dicho anteriormente, para que os imprima una cadena:



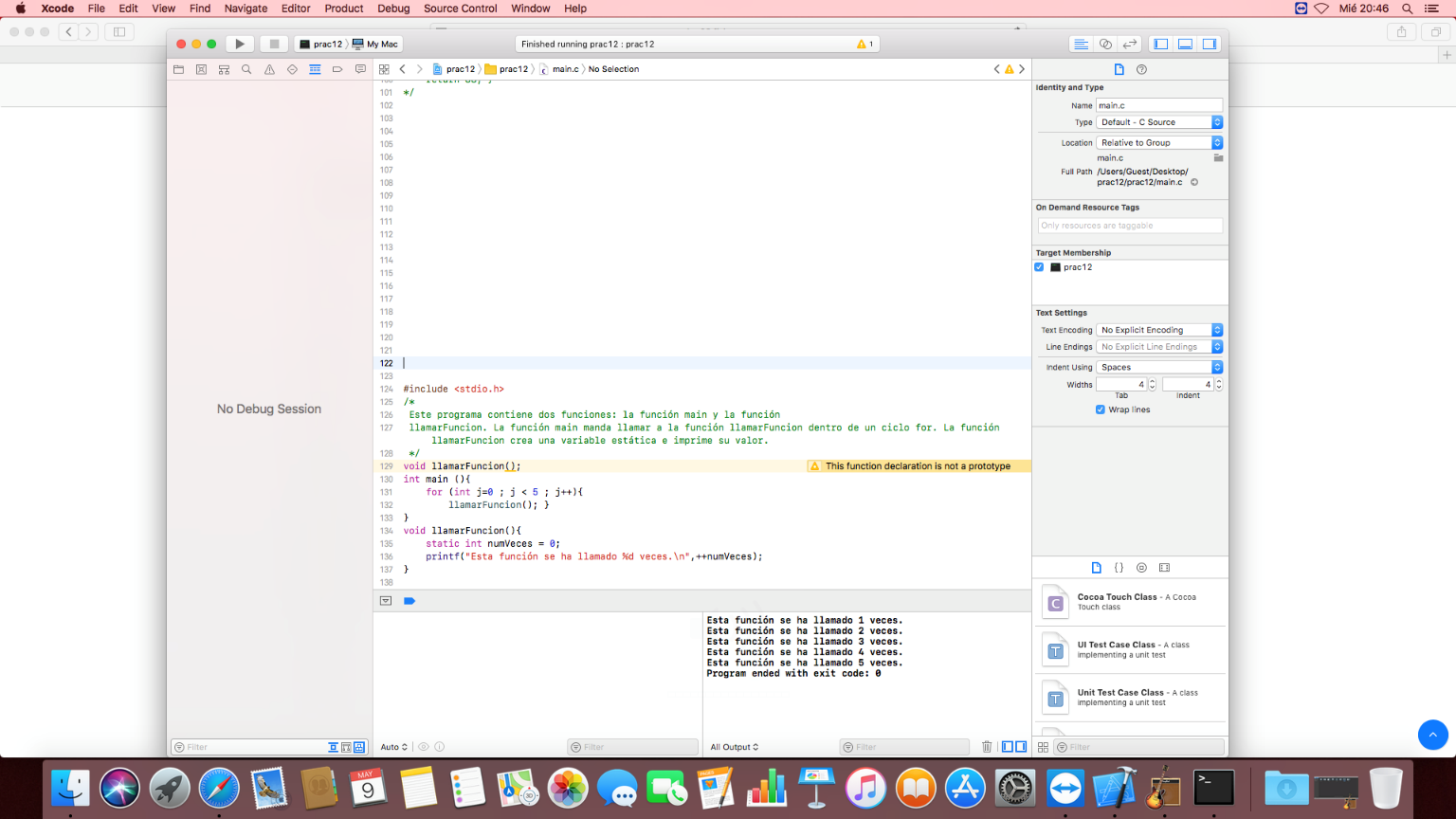
En el siguiente ejemplo vimos el uso de argumentos en main, por lo tanto tuvimos que tilizar la terminal para que obtuviéramos los resultados deseados, ya que los argumentos se tenían que dar durante la compilación:



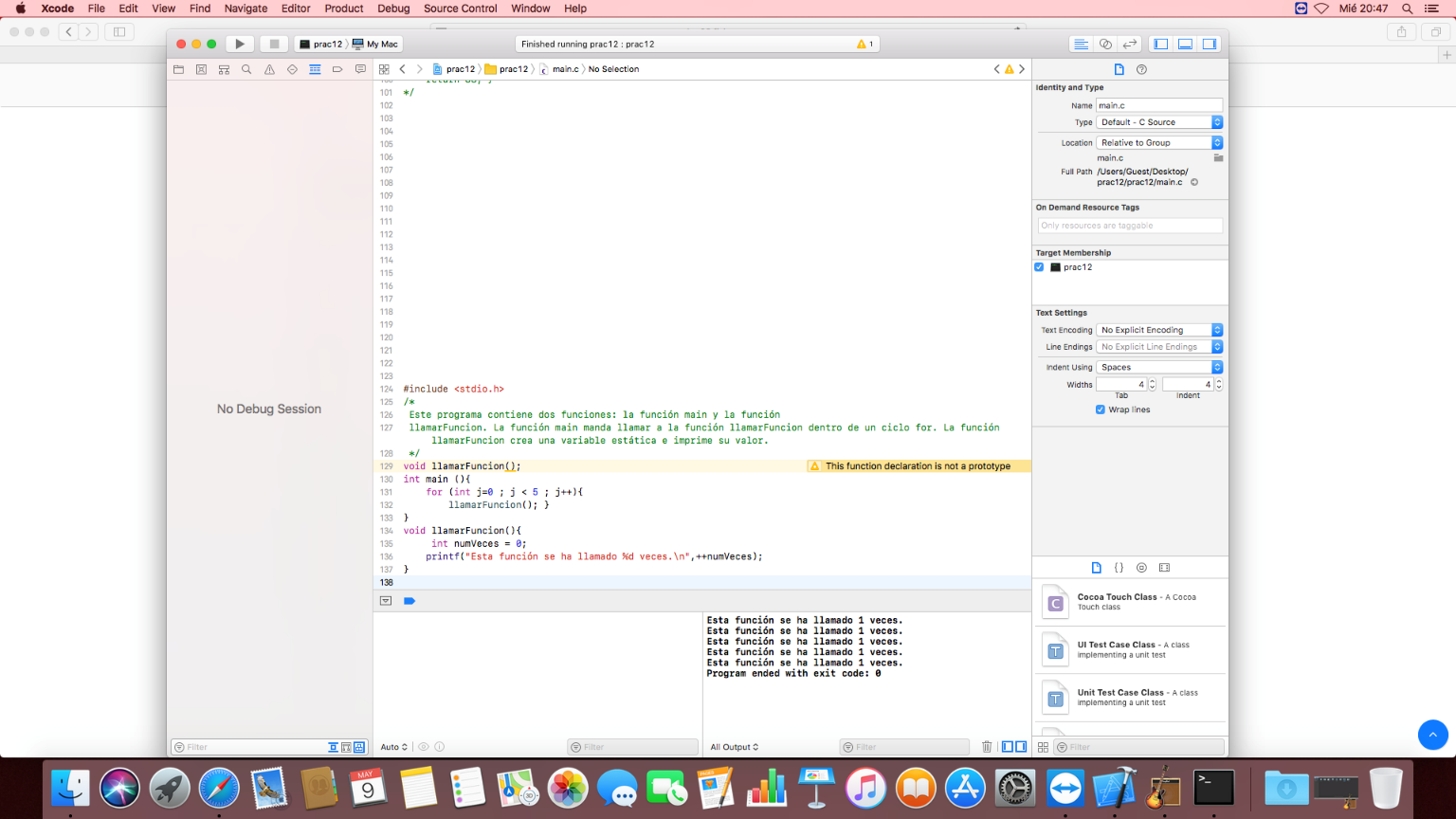




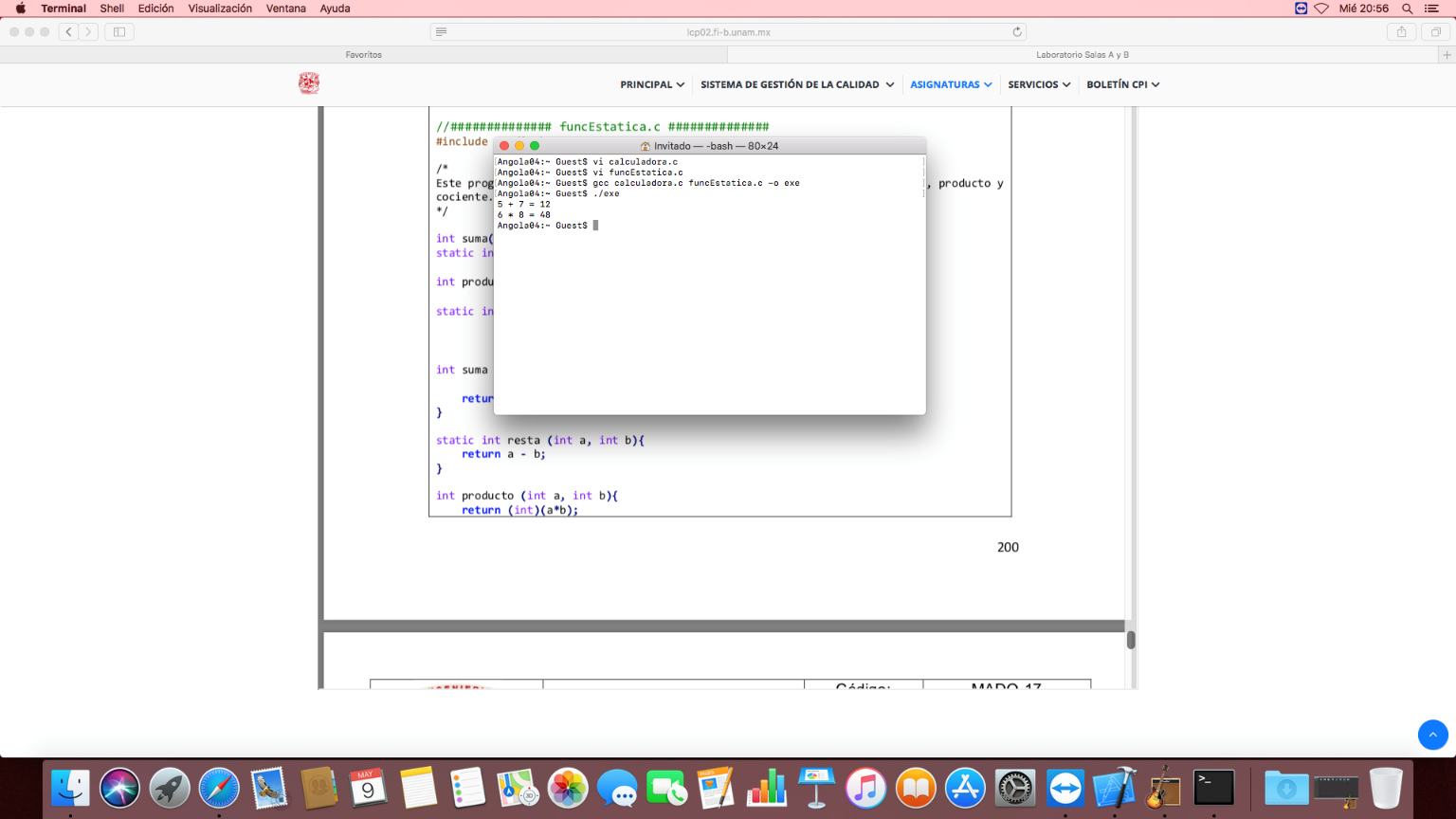
Ahora en el siguiente ejemplo que vimos, hacemos utilización de la static, que lo que hace es mantener cierto valor y evita que allá cambios en el:

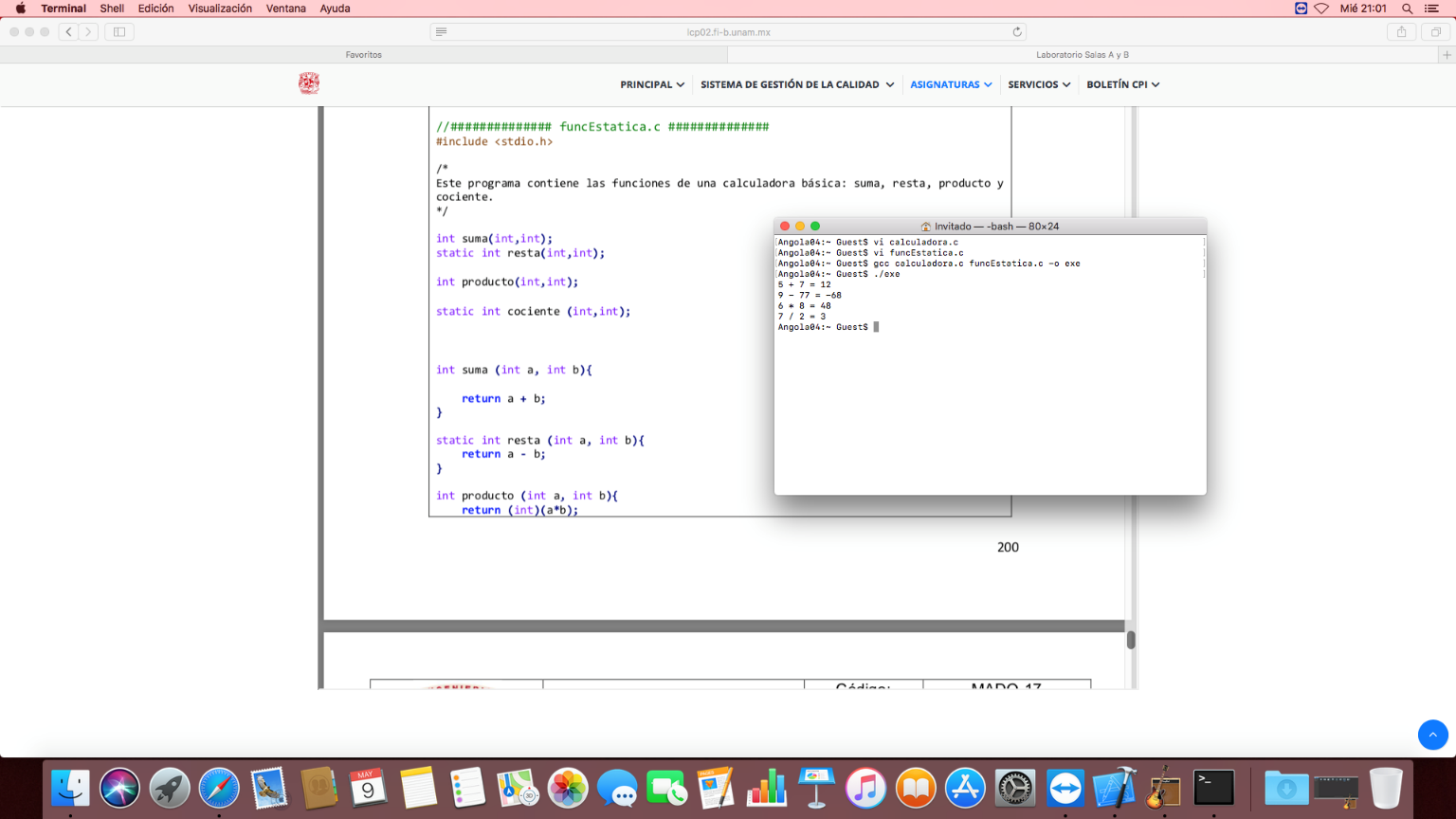


Y gracias a esto pude hacer la ejecución correctamente, ya que le quitamos el static y ahora no se hizo bien



Y por último vimos uso del estatic en funciones, entonces vimos el siguiente código y lo ejecutamos, y nos mostró solo dos valores por los static

Después le quitamos los static para ver que resulto y nos salió esto:



Ejercicios propuestos:

Ejercicios propuestos de práctica de funciones

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

* cuadrado de un número,
* el mayor de 2 números
* el menor de 2 números)

Esta es una idea del programa

**#include <stdio.h>**

**#include <conio.h>**

**float a, b;**

**char enter,pausa;**

**float cuadrado(int);**

**int mayor(int, int);**

**int menor(int, int);**

**void main( )**

**{**

**textbackground(BLUE); textcolor(YELLOW);**

**clrscr();**

**gotoxy(30,.2); printf(“ Práctica #12 “);**

**gotoxy(3,.4); printf(“ Ejemplos de varias funciones programadas y de biblioteca”);**

**gotoxy(10,.6); printf(“ Teclee un número: “);**

**scanf (“% f %c”,&a,&enter);**

**gotoxy(10,.7); printf(“ Teclee otro número: “);**

**scanf (“% f %c”,&b,&enter);**

**printf(“ \n\n El cuadrado de : %f es: %12.4f”, a, cuadrado (a));**

**printf(“ \n\n El mayor de los dos números es el: %d”, mayor (a,b));**

**printf(“ \n\n El menor de los dos números es el: %d”, menor (a,b));**

**printf(“ \n\n Presione <ENTER> para continuar...”);**

**scanf (“%c”,&pausa);**

**textbackground(BLACK); textcolor(WHITE);**

**getch();**

**}**

**float cuadrado (int a)**

**{**

**return (a\*a);**

**}**

**int mayor(int x, int y)**

**{**

**if (x>y)**

**return (x);**

**else**

**if(y >x)**

**return (y);**

**else**

**return (0);**

**}**

**int menor(int xx, int yy)**

**{**

**if (xx<yy)**

**return (xx);**

**else**

**if(y y<xx)**

**return (yy);**

**else**

**return (0);**

**}**

Debe mejorarlo o crear su propia solución

Dado el programa anterior, además de editarlo, debe guardarlo, compilarlo y ejecutarlo.

Describa:

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

Primero se borro los textbackgroun,textcolor, los goto, en el ultimo if había un espacio en “y y” que marcaba error y los printf el %f tambien tenia espacio, asi como el &enter era inecesario y en los resultados, igual quitar la opción de enter, el pausa, junto con el scnf del final, sus declaraciones, la librería conio y cambiar los valores de a,b,xx,yy por una x,y que era más cencillo

b) el desplegado obtenido

Lo que se obtiene es exactamente lo que se pide en el programa

Marque en el programa donde se realizan:

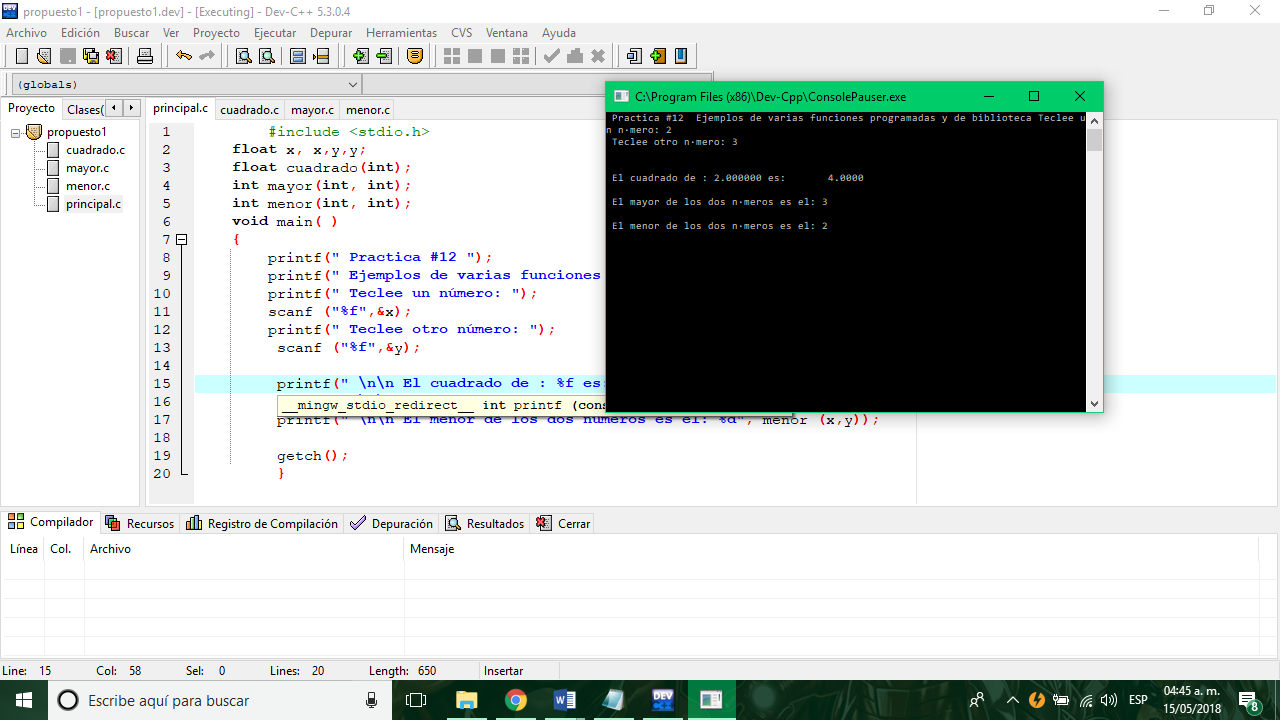
a) las llamadas a las funciones

b) la declaración de las funciones

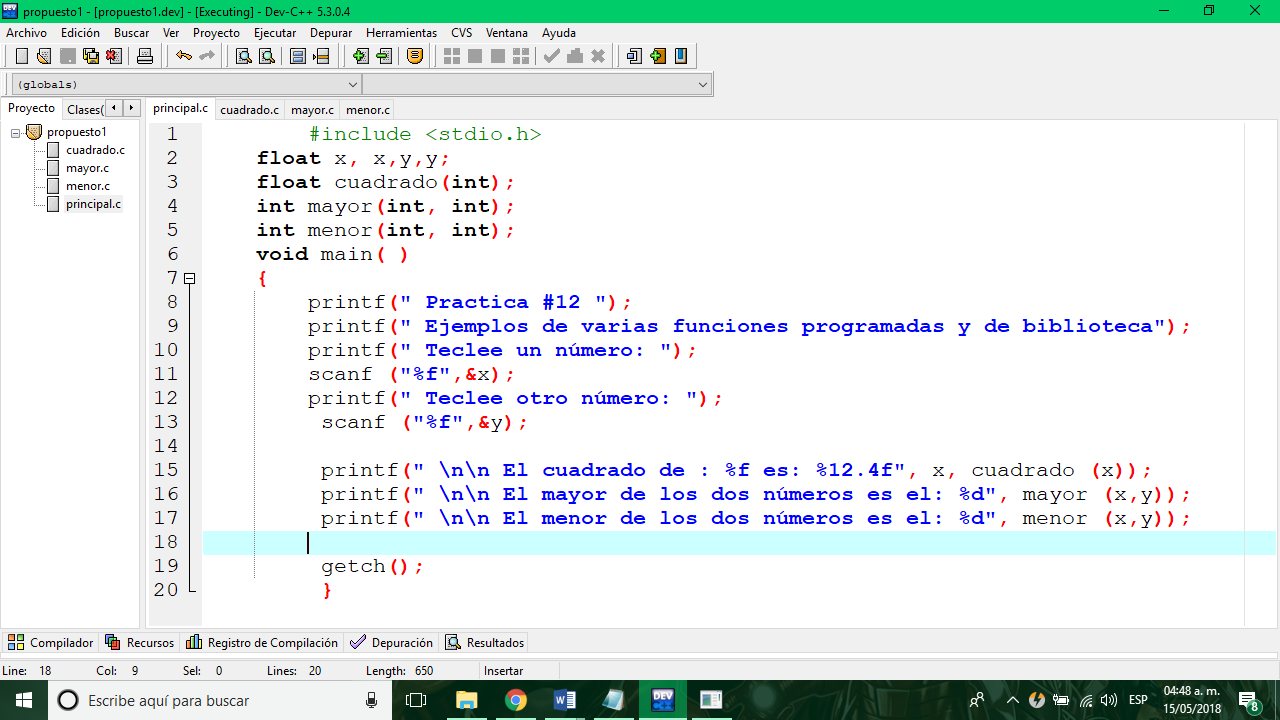
c) la definición de las funciones

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.

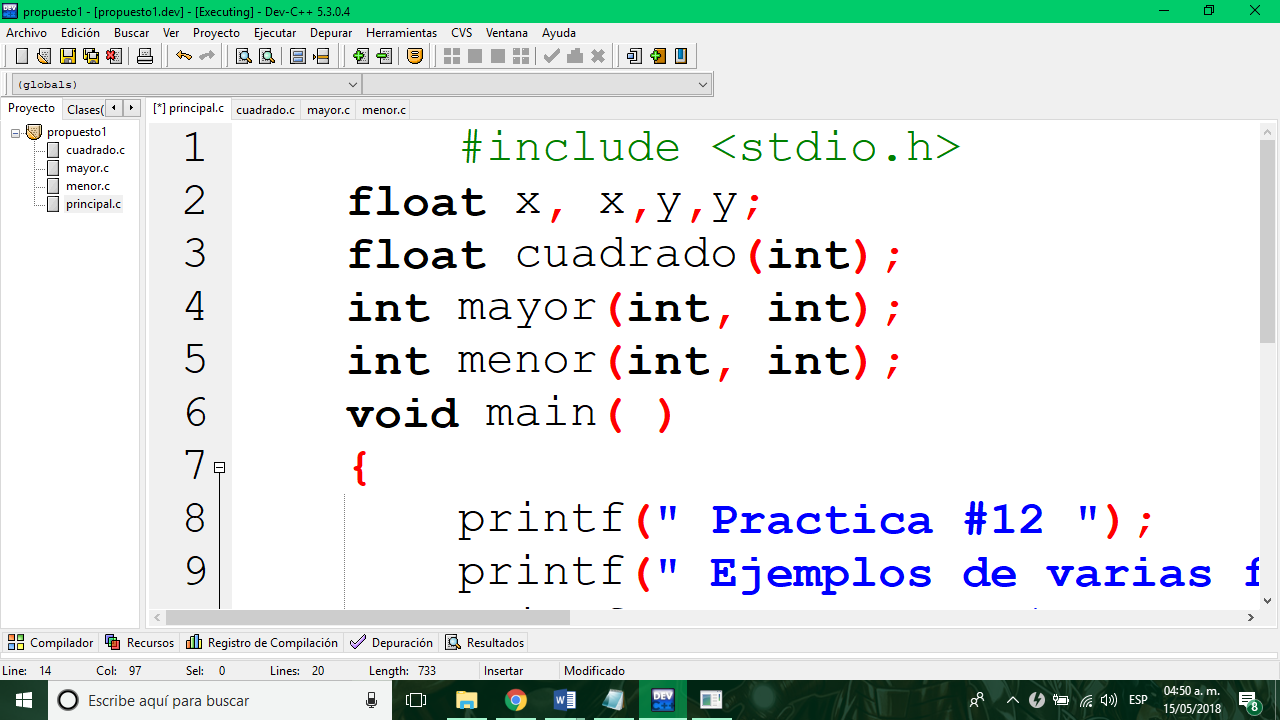
b) El desplegado



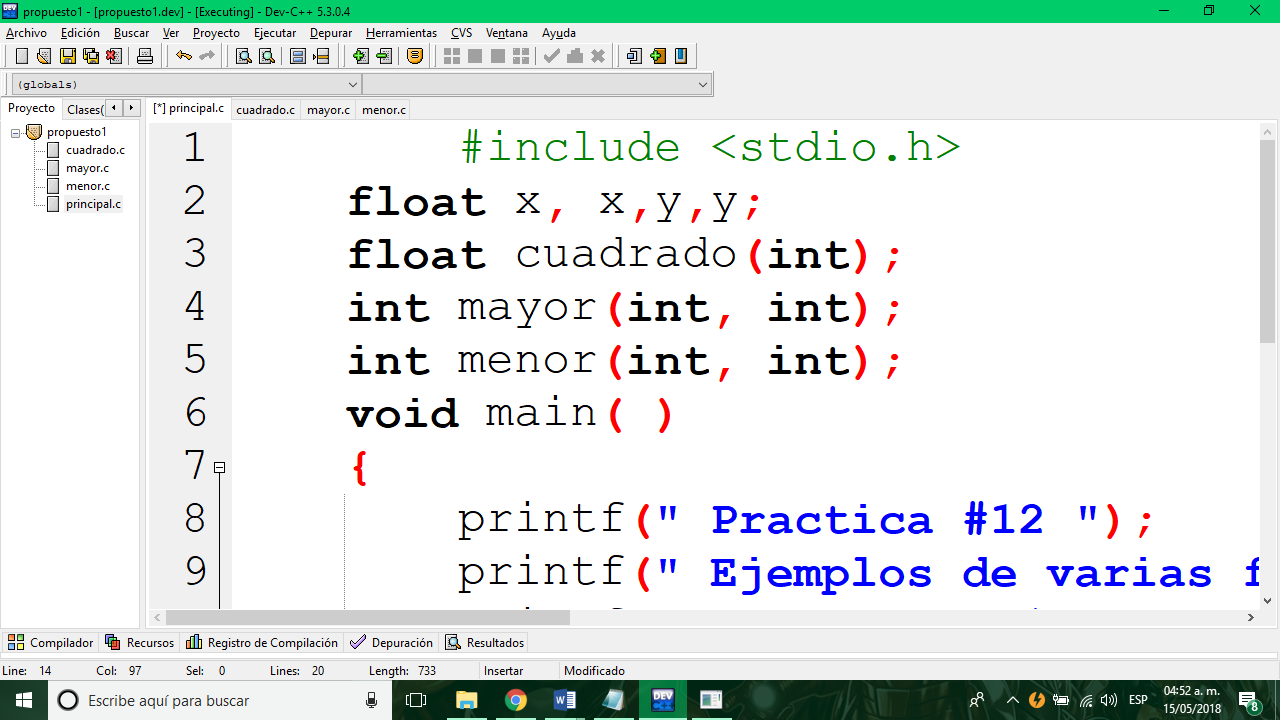
a) Las llamada a funcion: (sin el “%d” que se ve)



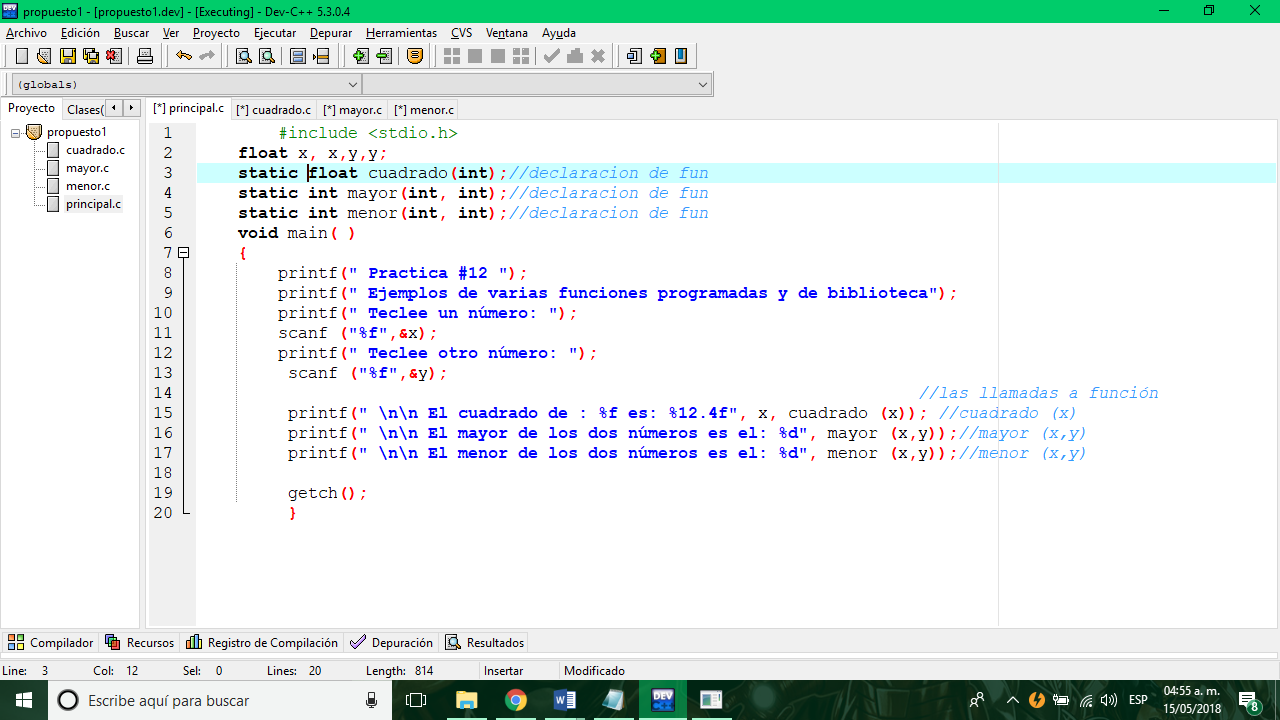
b) la declaración:



C) definición de las funciones: en un archivo dentro del projecto:



Al cambiar las declaraciones de fun por estáticas:



No pasa nada, el resultado es el mismo.

2.- Resolver el problema de ordenamiento de un arreglo que almacena 6 números usando apuntadores y funciones

Conclusión: Podemos concluir sabiendo el funcionamiento de static y del uso de funciones, correctamente, aunque ya lo habíamos visto antes, gracias a esta práctica y a los ejercicios, todo quedo más claro y esto nos ayudara en el futuro que necesitemos utilizar las grandiosas funciones.

Bibliografía:

* El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
* http://lcp02.fi-b.unam.mx/