



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

# Laboratorio de Computación Salas A y B

*Profesor:* Ing. Karina García Morales

*Asignatura:* Fundamentos de Programación

*Grupo:* 1121

*No de Práctica(s):* 12

*Integrante(s):* Jacinto Rodríguez Moisés Rodrigo

*No. de Equipo de  
cómputo empleado:* 6

*Semestre:* 2019-1

*Fecha de entrega:* 13/11/18

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Práctica: 12 Funciones

### 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.

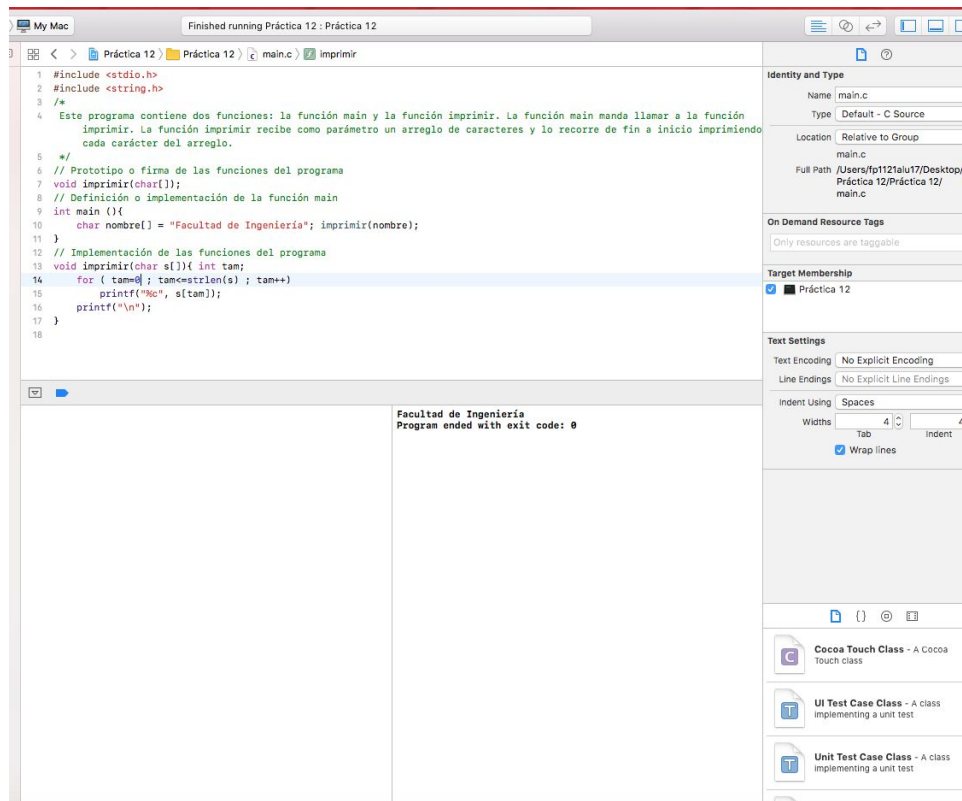
### Desarrollo:

Las funciones se declaran cuando esta abajo de la librería con un punto y coma al último.

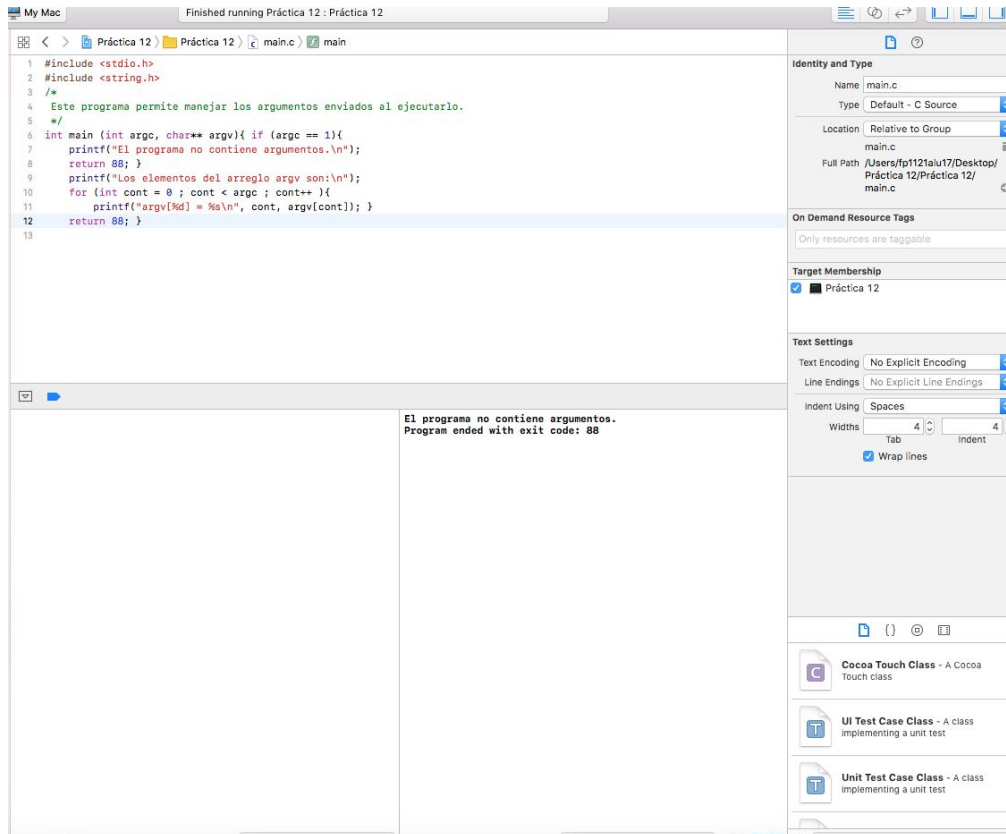
Cuando int main se pone al último del programa, la función se coloca debajo de la librería; no esta declarado

La firma de una función es cuando la declaramos en un inicio.

para el siguiente programa se editó el ciclo for para la correcta ejecución e impresión de dicho programa.



La función principal tiene parámetros dentro de los paréntesis que indica mediante iteración es de if si se cumple o no para mandarlo a imprimir al usuario.



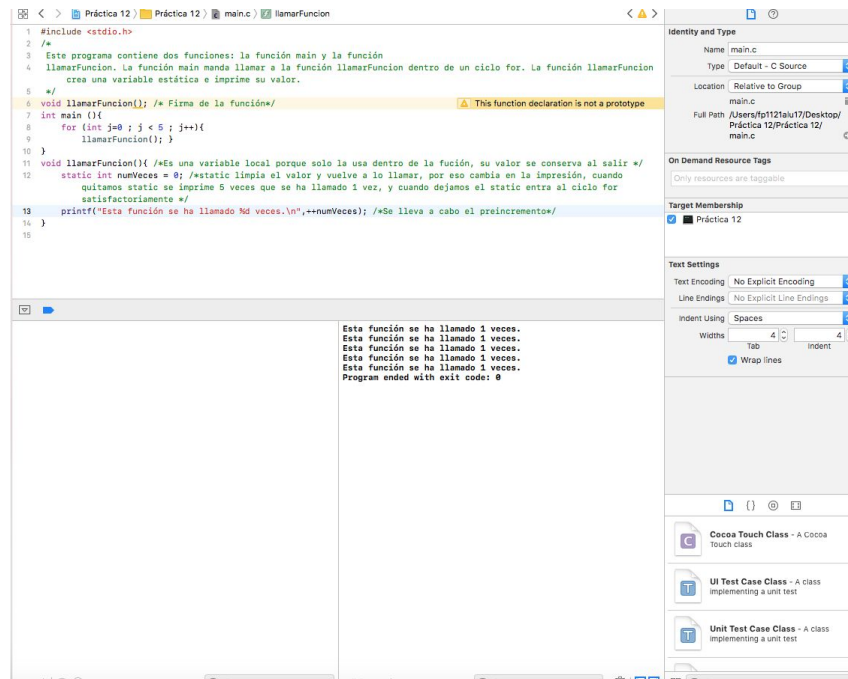
The screenshot shows an IDE window titled "My Mac" with a sub-header "Finished running Práctica 12: Práctica 12". The editor displays a C program in a file named "main.c". The code includes `<stdio.h>` and `<string.h>`, followed by a comment in Spanish. The `main` function takes `argc` and `argv` as parameters. It uses an `if` statement to check if `argc` is 1. If true, it prints "El programa no contiene argumentos." and returns 88. If false, it prints "Los elementos del arreglo argv son:" and enters a `for` loop that iterates over `argv` using a local variable `cont`, printing each element. The program ends with `return 88;`. The output window below the editor shows the execution result: "El programa no contiene argumentos." followed by "Program ended with exit code: 88". The right sidebar contains a "Properties" panel with "Identity and Type" (Name: main.c, Type: Default - C Source), "On Demand Resource Tags" (Only resources are taggable), "Target Membership" (checked for Práctica 12), and "Text Settings" (Text Encoding: No Explicit Encoding, Line Endings: No Explicit Line Endings, Indent Using: Spaces, Widths: 4, Tab: 4, Wrap lines: checked). At the bottom, a file explorer shows a project structure with "Cocoa Touch Class - A Cocoa Touch class" and two "Unit Test Case Class - A class implementing a unit test" files.

```
1 #include <stdio.h>
2 #include <string.h>
3 /*
4  Este programa permite manejar los argumentos enviados al ejecutarlo.
5  */
6 int main (int argc, char** argv){ if (argc == 1){
7     printf("El programa no contiene argumentos.\n");
8     return 88; }
9     printf("Los elementos del arreglo argv son:\n");
10    for (int cont = 0 ; cont < argc ; cont++){
11        printf("argv[%d] = %s\n", cont, argv[cont]); }
12    return 88; }
13
```

El programa no contiene argumentos.  
Program ended with exit code: 88

La variable `cont` es local, por lo que solo se puede usar dentro de la función, y cuando se manda a llamar desde fuera de la función simplemente no responde

Cuando hay una variable declarada como static accede a la unidad de memoria del dispositivo y no cambia su valor hasta el término del programa; solo se usa dentro de una función en la que fue declarada.



Cuando se compilan los dos archivos al mismo tiempo (gcc funcEstatica.c calculadora.c -o exe), las funciones suma y producto son accesibles desde el archivo calculadora y, por tanto, se genera el código ejecutable. Si se quitan los comentarios y se intenta compilar los archivos se enviará un error, debido a que las funciones son estáticas y no pueden ser accedidas fuera del archivo funcEstaticas.c.



```
fp1121alu14 — -bash — 80x24
Antartida05:~ fp1121alu14$ gcc funcEstatica.c calculadora.c -o exe
Antartida05:~ fp1121alu14$ vi funcEstatica.c
Antartida05:~ fp1121alu14$ vi calculadora.c
Antartida05:~ fp1121alu14$ vi funcEstatica.c
Antartida05:~ fp1121alu14$ vi funcEstatica.c
Antartida05:~ fp1121alu14$ vi funcEstatica.c
Antartida05:~ fp1121alu14$ ./ exe
-bash: ./: is a directory
Antartida05:~ fp1121alu14$ ./exe
5 + 7 = 12
Antartida05:~ fp1121alu14$ vi calculadora.c
Antartida05:~ fp1121alu14$ vi calculadora.c
Antartida05:~ fp1121alu14$ vi funcEstatica.c
Antartida05:~ fp1121alu14$ gcc funcEstatica.c calculadora.c -o exe
Antartida05:~ fp1121alu14$ ./exe
5 + 7 = 12
9 - 77 = -68
6 * 8 = 48
7 / 2 = 3
Antartida05:~ fp1121alu14$
```

## Tarea

Ejercicios propuestos de práctica de funciones

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

- cuadrado de un numero,
- el mayor de 2 números
- el menor de 2 números)

```
#include<stdio.h>
int funPotencia();
void mayor();
int menor();
int main(){
    float x;
    int valor;
    do {
        printf("\t\n  Selecciona la función que deseas ejecutar\n");
        printf("\n1.-Potencia al cuadrado de un número");
        printf("\n2.-El Mayor de dos números");
        printf("\n3.-El Menor de dos números");
        printf("\n4.-Salir\n");
        scanf("%d",&valor);
        switch(valor){
            case 1:
                x=funPotencia();
                printf("La potencia al cuadrado es: %f",x);
            break;
            case 2:
                mayor();
            break;
            case 3:
```

```

        menor();
    break;
    default:
        printf("\nsaliste del programa");
        break;
    }
}

while(valor>=1&&valor<4);

return 0;
}
//Función que obtiene la potencia al cuadrado de un número
int funPotencia()
{
    float valorX;
    printf("\nIntroduce el número al que le quieres aplicar potencia al cuadrado\n");
    scanf("\n%f",&valorX);
    return valorA * valorX;
}

```

//Función que obtiene el mayor de dos números dados por el usuario

```

void mayor ()
{
    int y1, y2;

    printf( "\n  Introduce un número entero: ");
    scanf( "%d", &y1 );
    printf( "\n  Introduce otro número entero: ");
    scanf( "%d", &y2 );

    if ( y1 > y2 )
        printf( "\n  El %d es mayor.", y1 );
    else

        if ( y1 < y2 )
            printf( "\n  El %d es mayor.", y2 );
        else
            printf( "\n  Son números iguales" );
}

```

//Función que obtiene el menor de dos números dados por el usuario

```

int menor (){
    int y1, y2;

    printf( "\n  Introduce un número entero: ");

```

```
scanf( "%d", &y1 );
printf( "\n  Introduce otro número entero: ");
scanf( "%d", &n2 );

if ( y1 < y2 )
    printf( "\n  El %d es menor.", y1 );
else

    if ( y1 > y2 )
        printf( "\n  El %d es menor.", y2 );
    else
        printf( "\n  Son números iguales" );
    return 0;
}
```