



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Alejandro Pimentel

*Asignatura:* Fundamentos de programación

*Grupo:* 3

*No de Práctica(s):* Practica 12

*Integrante(s):* Garcés Gallardo Julio César

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:* 14

*Semestre:* Primer semestre

*Fecha de entrega:* 4 de Noviembre del 2019

*Observaciones:* La segunda actividad también se debía implementar como una función. Al final, debías tener dos funciones independientes de main en donde una llamara a la otra (suma a factorial)

**CALIFICACIÓN:** 8

**Tema:** 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.

**Introducción:** Se crearán programas como se han visto anteriormente, con la única diferencia de que estas contarán con un “prototipo”, un prototipo es aquella función que se anuncia al principio del programa, pero se define al final de ella.

Este prototipo se puede usar a lo largo del programa y la computadora ya sabrá que existe dicha función.

**Desarrollo:**

Las actividades deben tener los prototipos de sus funciones, y sus funciones implementadas después del main.

- **Crear un programa que tenga una función que regrese la factorial de un número de entrada.**

Este es el programa que tiene como prototipo el “doublé factorial (long int)”:

```
#include <stdio.h>

double factorial(long int);

int main(){
    long int x, resultado, y;
    printf("Ingrese un valor: ");
    scanf("%li", &x);
    y=x;
    resultado=factorial(x);
    printf("El factorial de %li es %li.\n", y, resultado);
    return 0;
}

double factorial(long int x){
    long int resultado=1;
    while(x>1){
        resultado=resultado*x;
        x--;
    }
    return resultado;
}
```

Aquí compruebo el funcionamiento del programa en la terminal:

```

Peppermint Terminal
julio@julio-VirtualBox ~ $ cd Descargas/
julio@julio-VirtualBox ~/Descargas $ ls
act      act34      activ      main.c    P12.c     rs15      rs18.c    rs8.c
act2     act35      actividad1 p11       rs10      rs15.c    rs19      rs9.c
act2.c   act36      actividad1.c p11.1    rs10.c    rs16      rs19.c
act3     act37      actividad2  P112     rs11.c    rs16.c    rs5
act30    act38      actividad2.c P112.c   rs12.c    rs17      rs5.c
act31    act39      actividad3  P113     rs13.c    'rs17(1).c' rs6
act32    act3.c     actividad3.c P114     rs14      rs17.c    rs6.c
act33    acti       main       p11.c    rs14.c    rs18      rs7.c
julio@julio-VirtualBox ~/Descargas $ gcc P12.c -o P12
julio@julio-VirtualBox ~/Descargas $ ./P12
Ingrese un valor: 5
El factorial de 5 es 120.
julio@julio-VirtualBox ~/Descargas $ ./P12
Ingrese un valor: 12
El factorial de 12 es 479001600.
julio@julio-VirtualBox ~/Descargas $ ./P12
Ingrese un valor: 7
El factorial de 7 es 5040.
julio@julio-VirtualBox ~/Descargas $ _

```

- Crear un programa que tenga una función que regrese el resultado de la serie:

$$\sum_{x=1}^n \frac{x!}{x}$$

Para un número n de entrada. Utilizar la función de factorial de la primera actividad.

Este es el programa que cree:

```

#include <stdio.h>

double factorial(long int);

int main(){
    long int x, resultado=0, y;
    printf("Ingrese un valor: ");
    scanf("%li", &x);
    for(int i=0; i=x; i++){
        y=factorial(x);
        y=y/x;
        resultado=resultado+y;
        x--;
    }
    printf("%li\n", resultado);
    return 0;
}

double factorial(long int x){
    long int y=1;
    while(x>1){
        y=y*x;
        x--;
    }
    return y;
}

```

La suma también se debía de implementar como función

Esto es de cuando corrí el programa:

```
julio@julio-VirtualBox ~/Documentos $ gcc P12.c -o P127
julio@julio-VirtualBox ~/Documentos $ ./P127
Ingrese un valor: 3
4
julio@julio-VirtualBox ~/Documentos $ gcc P12.c -o P127
julio@julio-VirtualBox ~/Documentos $ ./P127
Ingrese un valor: 4
10
```

**Conclusión:** Concluyo que el uso de estos prototipos es útil ya que puedes empezar a hacer tu programa y dividirlo en dos partes y en mi caso esto se volvió más fácil, pero obviamente hubo algunos casos que me confundieron porque no sabía dónde se encontraba el error.