



## Laboratorios de computación Salas A y B.

<i>Profesor:</i>	Alejandro Esteban Pimentel Alarcon
<i>Asignatura :</i>	Fundamentos de programación
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	12
<i>Integrante(s):</i>	Flores Rodriguez Ricardo Manuel 317189795 Valencia Moya José Alejandro 317098543
<i>No. de Equipo de cómputo empleado:</i>	51 y 52
<i>No. de Lista o Brigada:</i>	12 y 51
<i>Semestre:</i>	Primero 2019-2020
<i>Fecha de entrega:</i>	04 de noviembre del 2019
<i>Observaciones:</i>	Muy bien

**CALIFICACIÓN:** 10

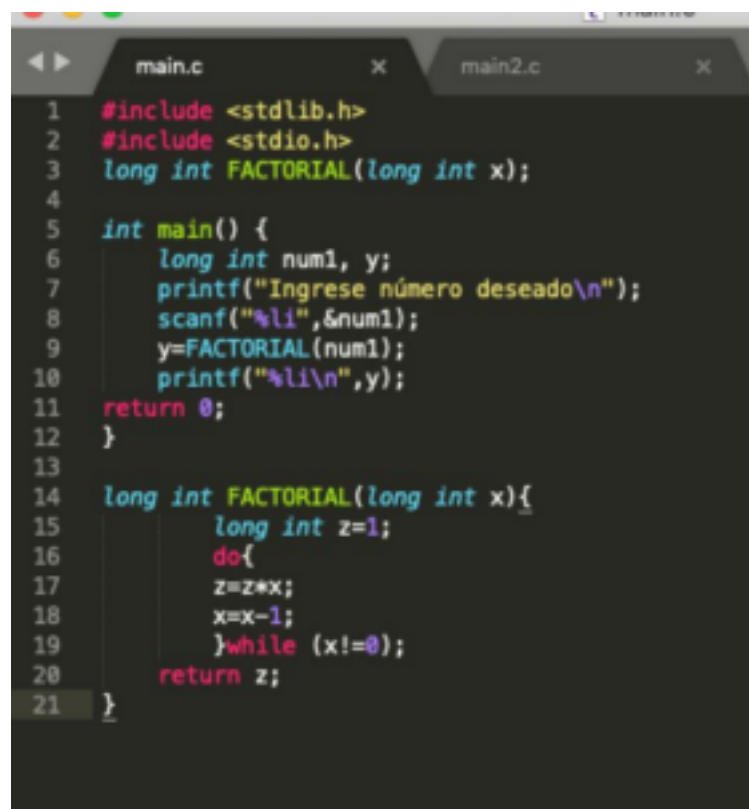
## Introducción

Cuando una acción se quiere repetir en diversas ocasiones dentro de un código se usan las funciones. Dichas funciones deben ir fuera de `int main` para que al momento de elaborar el código solo se llamen y se realice la operación programada en dicha función. Sin embargo, no es necesario que vayan antes del `int main`, ya que se pueden utilizar simplemente firmas o prototipos de una función antes de `int main` y poner el resto de la función al final del código. A lo largo del presente documento se mostrará la diferencia entre una firma de función y su implementación, la importancia de usar tanto a las funciones como a su firma y la manera correcta de elaborar funciones en el programa C.

## 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.

**Actividad 1:** Crear un programa que tenga una función que regrese el factorial de un número de entrada.



```
1  #include <stdlib.h>
2  #include <stdio.h>
3  long int FACTORIAL(long int x);
4
5  int main() {
6      long int num1, y;
7      printf("Ingrese número deseado\n");
8      scanf("%li", &num1);
9      y=FACTORIAL(num1);
10     printf("%li\n", y);
11     return 0;
12 }
13
14 long int FACTORIAL(long int x){
15     long int z=1;
16     do{
17         z=z*x;
18         x=x-1;
19     }while (x!=0);
20     return z;
21 }
```

```
Lituania09:desktop fp03alu08$ gcc main.c -o main
Lituania09:desktop fp03alu08$ ./main
Ingrese número deseado
3
6
Lituania09:desktop fp03alu08$ ./main
Ingrese número deseado
5
120
Lituania09:desktop fp03alu08$ ./main
Ingrese número deseado
10
3628800
Lituania09:desktop fp03alu08$ █
```

**Actividad 2:** Crear un programa que tenga una función que regrese el

resultado de la serie:  $\sum_{x=1}^n \frac{x!}{x}$

```

1  #include <stdlib.h>
2  #include <stdio.h>
3  long int FACTORIAL(long int x);
4  long int SERIE(long int x);
5
6  int main() {
7      long int num1, res;
8      printf("Ingrese número deseado\n");
9      scanf("%li", &num1);
10     res=SERIE(num1);
11     printf("%li\n", res);
12     return 0;
13 }
14
15 long int FACTORIAL(long int x){
16     long int z=1;
17     do{
18         z=z*x;
19         x=x-1;
20     }while (x!=0);
21     return z;
22 }
23
24 long int SERIE(long int x){
25     long int y;
26     long int s=1;
27     long int w=0;
28     do{
29         y=FACTORIAL(s)/s;
30         w=w+y;
31         s++;
32     }while (s<=x);
33     return w;
34 }

```

```

Session Editor View Macros Run Settings Ayuda
Ingrese número deseado
4
10
alumno@pcf:~$ ./main
Ingrese número deseado
10
409114
alumno@pcf:~$ ./main
Ingrese número deseado
3
4
alumno@pcf:~$ ./main 1
Ingrese número deseado
1
1
alumno@pcf:~$

```

## **Conclusiones**

Usar funciones dentro de los códigos computacionales permite no tener que volver a escribir la misma operación o acción a realizar, sino que simplemente llamar a la función programada con anterioridad. Por otra parte, usar la firma o prototipo de función es muy útil para tener un código mucho más organizado. Lo que hace mucho más práctico saber que funciones se tienen y cuales conviene usar.