	<h2 style="text-align: center;">Carátula para entrega de prácticas</h2>	
<p style="text-align: center;">Facultad de Ingeniería</p>	<p style="text-align: center;">Laboratorio de docencia</p>	

# Laboratorios de computación salas A y B

---

<i>Profesor:</i>	ALEJANDRO ESTEBAN PIMENTEL ALARCON
<i>Asignatura:</i>	FUNDAMENTOS DE PROGRAMACION
<i>Grupo:</i>	03
<i>No de Práctica(s):</i>	PRACTICA No. 12
<i>Integrante(s):</i>	MORONES LORES INGRID YOHUALLI
<i>No. de Equipo de cómputo empleado:</i>	
<i>No. de Lista o Brigada:</i>	
<i>Semestre:</i>	2020-1
<i>Fecha de entrega:</i>	4/NOVIEMBRE/2019
<i>Observaciones:</i>	
<i>CALIFICACIÓN:</i>	

# 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.

## INTRODUCCION.

En C, se conocen como funciones aquellos trozos de códigos utilizados para dividir un programa con el objetivo que, cada bloque realice una tarea determinada.

En las funciones juegan un papel muy importante las variables, ya que como se ha dicho estas pueden ser locales o globales.

**Variables Globales:** Estas se crean durante toda la ejecución del programa, y son globales, ya que pueden ser llamadas, leídas, modificadas, etc; desde cualquier función. Se definen antes del main().

**Variables Locales:** Estas, pueden ser utilizadas únicamente en la función que hayan sido declaradas.

La sintaxis de una función es la siguiente:

```
Tipo_de_datos nombre_de_la_funcion(tipo y nombre de argumentos)  
{  
    acciones  
}
```

Por lo tanto un programa en **c** debe constar de una o más **funciones**, y por supuesto no puede faltar la función principal main().

## ACTIVIDAD 1.

Crear un programa que tenga una función que regrese el factorial de un número de entrada.

C:\cygwin64\home\hugo morones\act1.c - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

```
act1.c x
1  #include <stdio.h>
2  long int FAC(int);
3
4  int main(){
5      long int n, x;
6      printf("Dame un numero\n");
7      scanf("%li", &n);
8      x=FAC(n);
9      printf("La factorial de tu numero es:\t %li \n", x);
10     return 0;
11 }
12 long int FAC(int n){
13     long int resultado;
14     resultado = 1;
15     while(n>0){
16         resultado=n*resultado;
17         n--;
18     }
19     return resultado;
20 }
```

Line 20, Column 2



Escribe aquí para buscar






```
hugo morones@LAPTOP-FGPEUFLH ~  
$ gcc act1.c -o prac  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./prac  
Dame un numero  
5  
La factorial de tu numero es:    120  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./prac  
Dame un numero  
4  
La factorial de tu numero es:    24  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./prac  
Dame un numero  
8  
La factorial de tu numero es:    40320  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ |
```

## ACTIVIDAD 2.

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.

 C:\cygwin64\home\hugo morones\act2.c - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

```

1  #include <stdio.h>
2  long int FAC(int);
3  long int SUMATORIA(int);
4  int main(){
5      int n;
6      int serie;
7      printf("Dame un numero\n");
8      scanf("%i", &n);
9      serie=SUMATORIA(n);
10     printf("El resultado de la serie es:\t %li \n", serie);
11     return 0;
12 }
13 long int FAC(int n){
14     int resultado;
15     resultado = 1;
16     while(n>0){
17         resultado=n*resultado;
18         n--;
19     }
20     return resultado;
21 }
22 long int SUMATORIA(int n){
23     int x,y;
24     int total;
25     total=0;
26     while(n>0){
27         x=FAC(n);
28         total=total+(x/n);
29         n--;
30     }
31     return total;
32 }

```

Line 32, Column 2



Escribe aquí para buscar





```
hugo morones@LAPTOP-FGPEUFLH ~  
$ gcc act2.c -o prac  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./prac  
Dame un numero  
5  
El resultado de la serie es:      34  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./prac  
Dame un numero  
6  
El resultado de la serie es:      154  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./prac  
Dame un numero  
8  
El resultado de la serie es:      5914  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ ./prac  
Dame un numero  
7  
El resultado de la serie es:      874  
  
hugo morones@LAPTOP-FGPEUFLH ~  
$ |
```

**CONCLUIONES.**

Se logró el objetivo resolvimos el problema dividiendo la solución en funciones.