



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Alejandro Esteban Pimentel Alarcon.

*Asignatura:* Fundamentos de programación

*Grupo:* 35

*No de Práctica(s):* 12

*Integrante(s):* Páez Martínez Karen

*No. de Equipo de  
cómputo empleado:* 2

*No. de Lista o Brigada:* 3781

*Semestre:* 1

*Fecha de entrega:* 04/11/2019

*Observaciones:* Muy bien

**CALIFICACIÓN:** 10

# PRACTICA#12

## INTRODUCCION:

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.

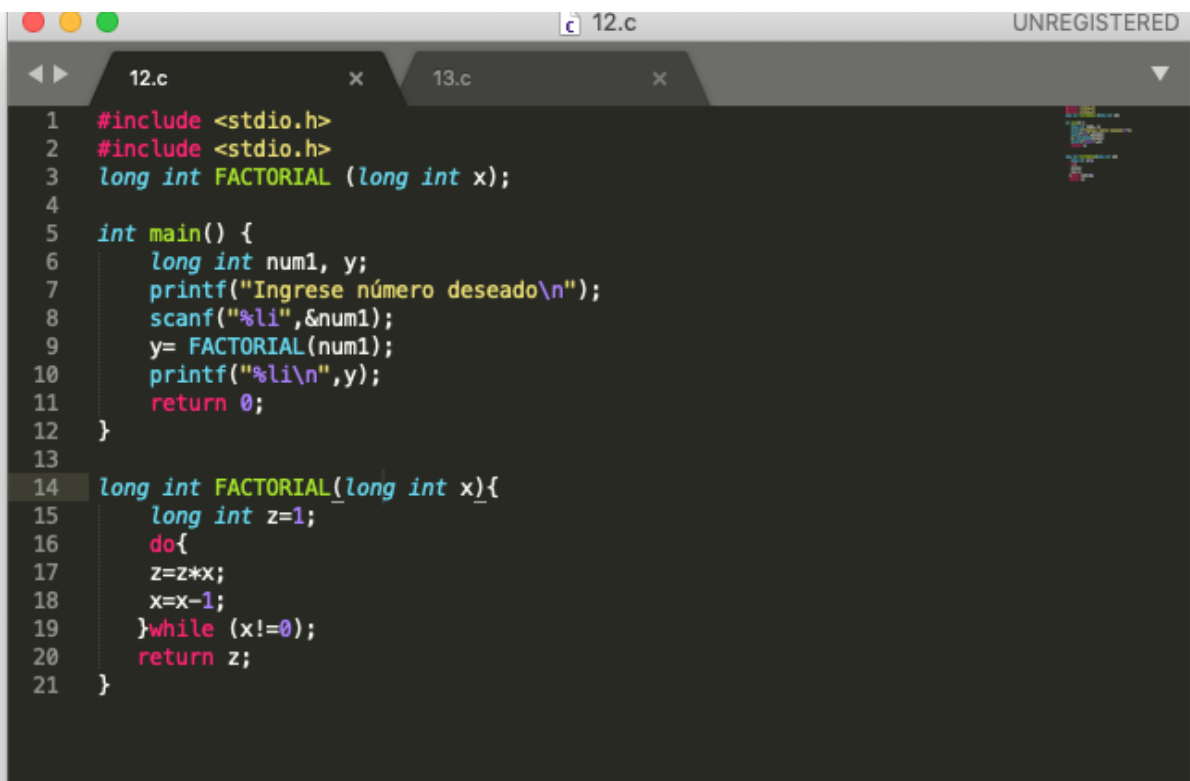
## OBJETIVO:

Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos.

## ACTIVIDAD#1

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 el factorial de un número de entrada.

1. Primero empesamos a programar y lo vamos desarrollando en sublime text.



The screenshot shows a code editor with two tabs: '12.c' and '13.c'. The '12.c' tab is active and displays the following C code:

```
1  #include <stdio.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 }
```

2. Después de esto empezamos a verificar si si corre o no en este caso corrio bien y se muestra el resultado que se pide con anterioridad.

```
Documents — -bash — 80x24

C99 [-Wimplicit-function-declaration]
y=FACTORIAL(num1);
^
12.c:14:20: error: unknown type name 'log'; did you mean 'long'?
long int FACTORIAL(log int x){
                    ^~~
                    long
12.c:14:10: error: conflicting types for 'FACTORIAL'
long int FACTORIAL(log int x){
^
12.c:9:4: note: previous implicit declaration is here
y=FACTORIAL(num1);
^
2 warnings and 3 errors generated.
[Lesoto04:Documents fp03alu40$ gcc 12.c -o 12
[Lesoto04:Documents fp03alu40$ ./12
Ingrese número deseado
8
40320
[Lesoto04:Documents fp03alu40$ ./12
Ingrese número deseado
5
120
Lesoto04:Documents fp03alu40$
```

## 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.

1. En esta actividad de igual manera empezamos a realizar en programa.

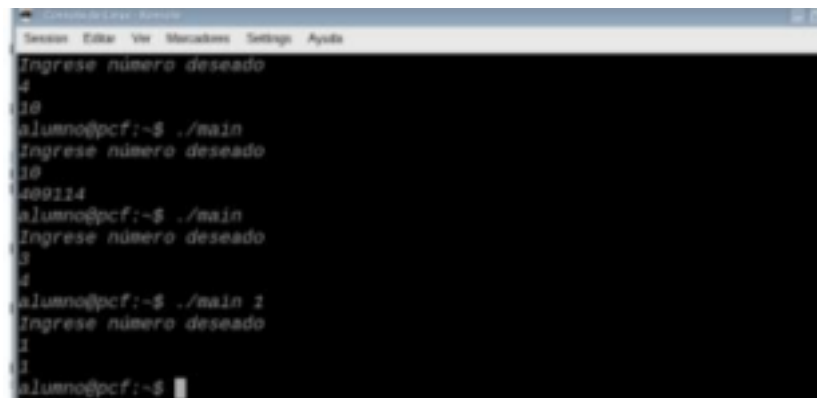
```
12.c 13.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 long int FACTORIAL(long int x);
4 long int SERIE(long int x);
5
6 int main() {
7     long int num1,res;
8     printf("Ingrese número deseado\n");
9     scanf("%li",&num1);
10    res=SERIE(num1);
11    printf("%li\n", res);
12    return 0;
13 }
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{
```

```

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 }

```

2. Aqui empezamos a comprobar si funciona o no el programa y que errors hay, en este caso si corrió bien y se muestra en la siguiente imagen.



```

Ingrese número deseado
4
24
alumno@pcf:~$ ./main
Ingrese número deseado
10
3628800
alumno@pcf:~$ ./main
Ingrese número deseado
3
6
alumno@pcf:~$ ./main 1
1
alumno@pcf:~$

```

## CONCLUSION:

En esta práctica pudimos ver la importancia de como programar y ver como componer factoriales, que en lo personal me costo mucho por que casi no encontraba como formular lo de factorial y hasta que el profe nos explico y ya me salio. Se puede decir que en esta practica se usa las 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. En lugar de tener en la parte de arriba del código mucho texto, solo se verán las funciones que se tienen y en la parte de abajo estará todo su desarrollo, lo que hace mucho más práctico saber que funciones se tienen y cuales convienen usar.