



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación Salas A y B

Profesor: Alejandro Pimentel

Asignatura: Laboratorio de protraction

Grupo: 135

No de Práctica(s): Práctica 12

Areli González Segura

Integrante(s):

*No. de Equipo de
cómputo empleado:* 23

No. de Lista o Brigada: 5319 no. Lista: 19

Semestre: Primer semestre

Fecha de entrega: 4/noviembre/2019

Observaciones:

CALIFICACIÓN: _____

Introducción:

Las funciones con retorno son muy parecidas a las funciones sin retorno pero su diferencia radica en para qué se utilizan.

Las funciones con retorno crean un “mini-programa” que se encarga de obtener, preparar o modificar una variable. La función con retorno nos devuelve ese valor justo cuando se lo pidamos, sin tenerle que decir una y otra vez al programa como obtener ese dato que necesitamos.

Las variables con las que jugamos en nuestras **funciones con retorno** pueden ser de cualquier tipo: numéricas, de texto o incluso vectores.

Objetivo:

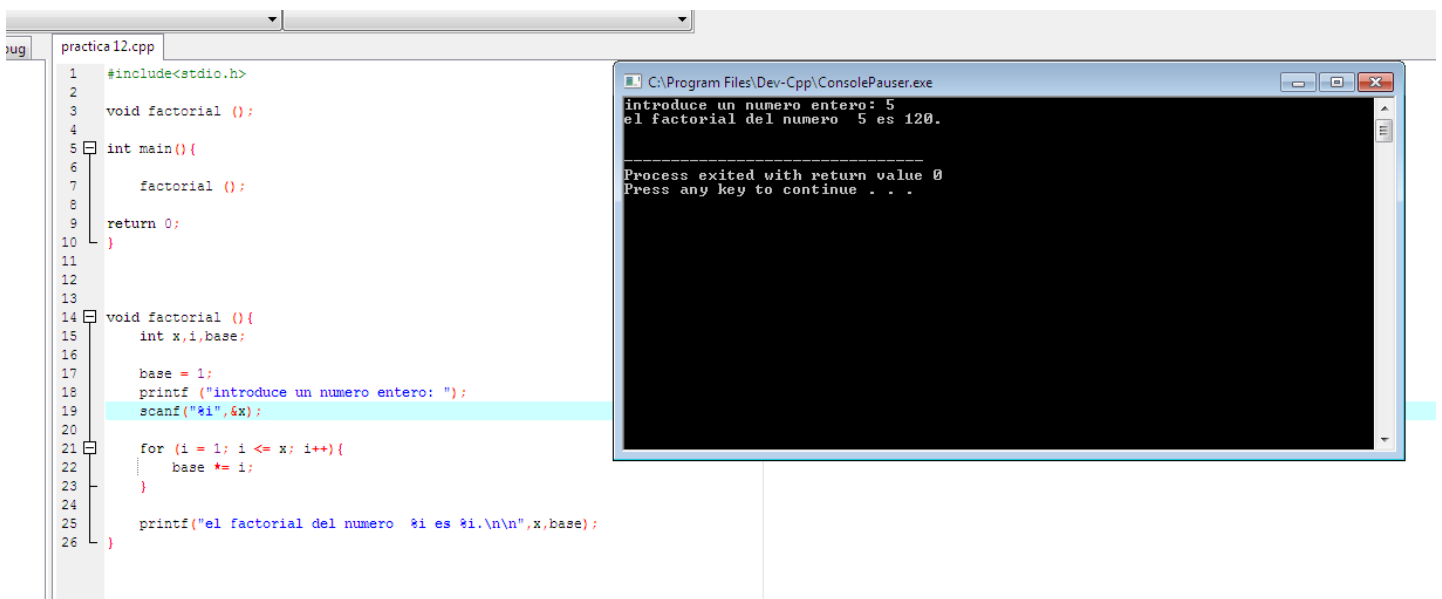
Elaborar programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

Desarrollo:

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

Actividad 1:

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



The image shows a code editor window on the left and a console window on the right. The code editor displays the source code for a C program named 'practica12.cpp'. The code defines a 'factorial' function and a 'main' function. The 'main' function calls 'factorial' and prints the result. The console window shows the output of the program, which prompts the user to enter a number, receives '5', and outputs 'el factorial del numero 5 es 120.'.

```
1 #include<stdio.h>
2
3 void factorial ();
4
5 int main(){
6
7     factorial ();
8
9     return 0;
10 }
11
12
13
14 void factorial (){
15     int x,i,base;
16
17     base = 1;
18     printf ("introduce un numero entero: ");
19     scanf("%i",&x);
20
21     for (i = 1; i <= x; i++){
22         base *= i;
23     }
24
25     printf("el factorial del numero %i es %i.\n\n",x,base);
26 }
```

```
C:\Program Files\Dev-Cpp\ConsolePauser.exe
introduce un numero entero: 5
el factorial del numero 5 es 120.

-----
Process exited with return value 0
Press any key to continue . . .
```

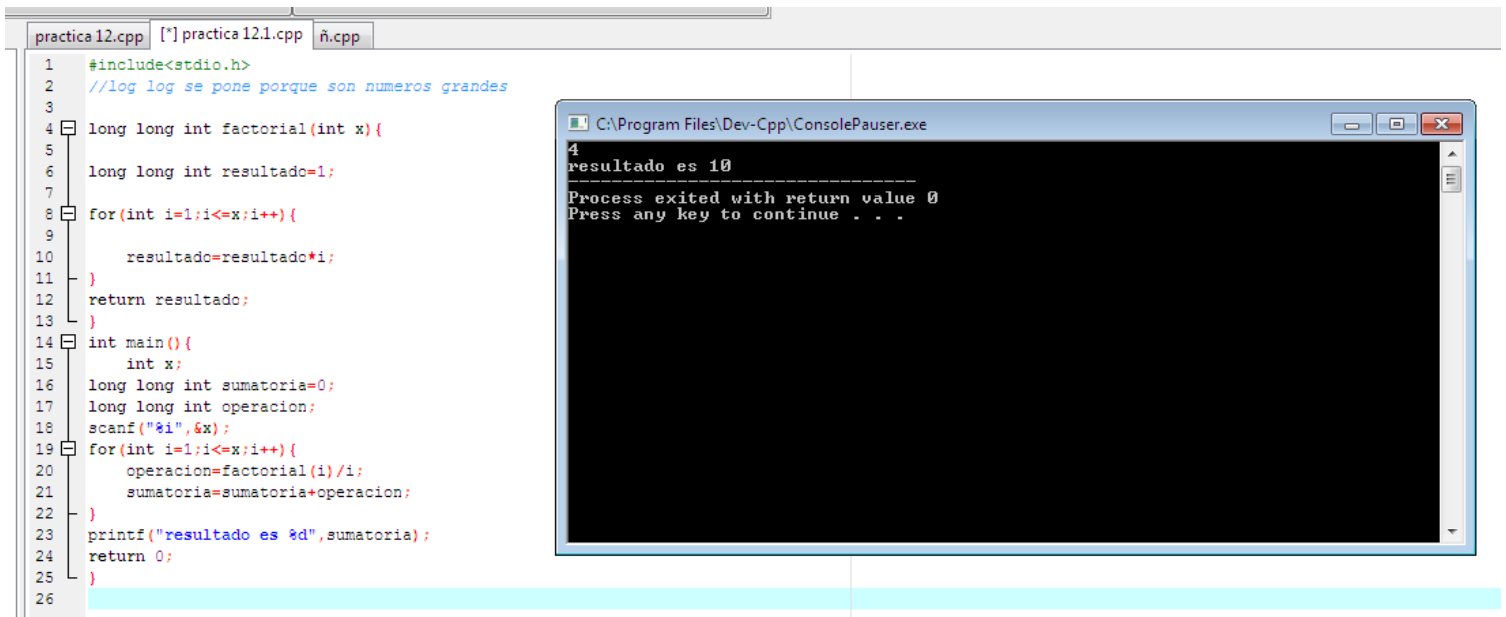
Actividad 2:

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

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

Nota:

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



```
practica 12.cpp | [*] practica 12.1.cpp | n.cpp
1  #include<stdio.h>
2  //log log se pone porque son numeros grandes
3
4  long long int factorial(int x){
5
6      long long int resultado=1;
7
8      for(int i=1;i<=x;i++){
9
10         resultado=resultado*i;
11     }
12     return resultado;
13 }
14 int main(){
15     int x;
16     long long int sumatoria=0;
17     long long int operacion;
18     scanf("%i",&x);
19     for(int i=1;i<=x;i++){
20         operacion=factorial(i)/i;
21         sumatoria=sumatoria+operacion;
22     }
23     printf("resultado es %d",sumatoria);
24     return 0;
25 }
26
```

```
C:\Program Files\Dev-Cpp\ConsolePauser.exe
4
resultado es 10
-----
Process exited with return value 0
Press any key to continue . . .
```

Conclusiones:

Estas funciones son muy útiles para que el programa sea más dinámico es decir, que nos evita que nuestro programa tenga mucho código y que sea más digerible, por ello considero que es bastante útil saber cómo ocuparlos puesto que es una herramienta que nos ahorra mucho trabajo cuando quieres hacer un programa que sea muy laborioso.