



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: MC. Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): Práctica 2

Integrante(s): Martínez Marcelino Dalila

*No. de Equipo de
cómputo empleado:* No. de cuenta: 313080119

No. de Lista o Brigada:

Semestre: 2020-1

Fecha de entrega: Lunes 4 de noviembre de 2019

Observaciones: Muy bien

CALIFICACIÓN: 10

Práctica No. 12

Introducción:

En esta práctica se hará uso de funciones en la resolución de problemas en programas de lenguaje C. Para ello, el estudiante conocerá primeramente a que se refiere el prototipo o firma de funciones, esto con el fin de establecerlas antes del programa principal. Después estructurara el programa principal llamando a las funciones donde se requiera su ejecución. Por último, desarrollara las funciones que el programa necesite, las cuales serán desarrolladas después del programa principal. En estas funciones se usará parámetros, de los cuales el estudiante podrá manipular.

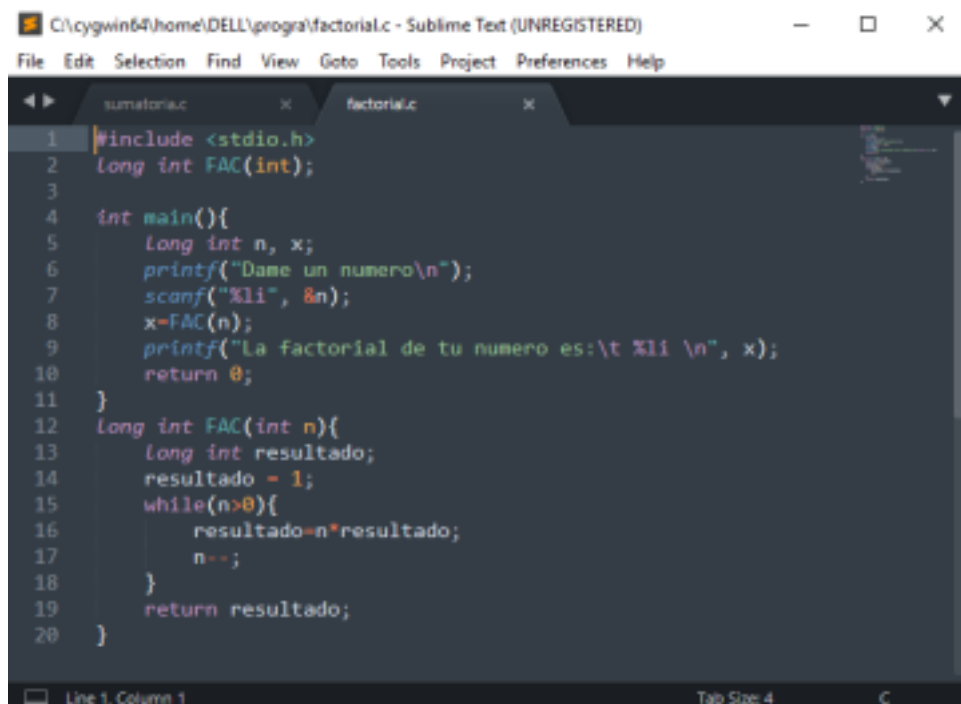
Objetivo:

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.

Actividades.

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.

A screenshot of a Sublime Text editor window titled "C:\cygwin64\home\DELL\progra\factorial.c - Sublime Text (UNREGISTERED)". The editor shows a C program with two tabs: "sumatoria.c" and "factorial.c". The code in "factorial.c" is as follows:

```
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 }
```

The status bar at the bottom indicates "Line 1, Column 1", "Tab Size: 4", and "C".

En este programa se utiliza una función que devuelve como resultado la factorial de un número

que el usuario ingresa, el cual será el parámetro para la función. Primero se hace el prototipo de la función que se va a utilizar en el programa principal, y después del main se estructura la función. En la función principal se llama a la función de la factorial y se guarda lo que este devuelve en una variable, para después imprimirla.

```
DELL@DESKTOP-NJV5QTU ~/progra
$ gcc factorial.c -o fac

DELL@DESKTOP-NJV5QTU ~/progra
$ ./fac
Dame un numero
5
La factorial de tu numero es: 120

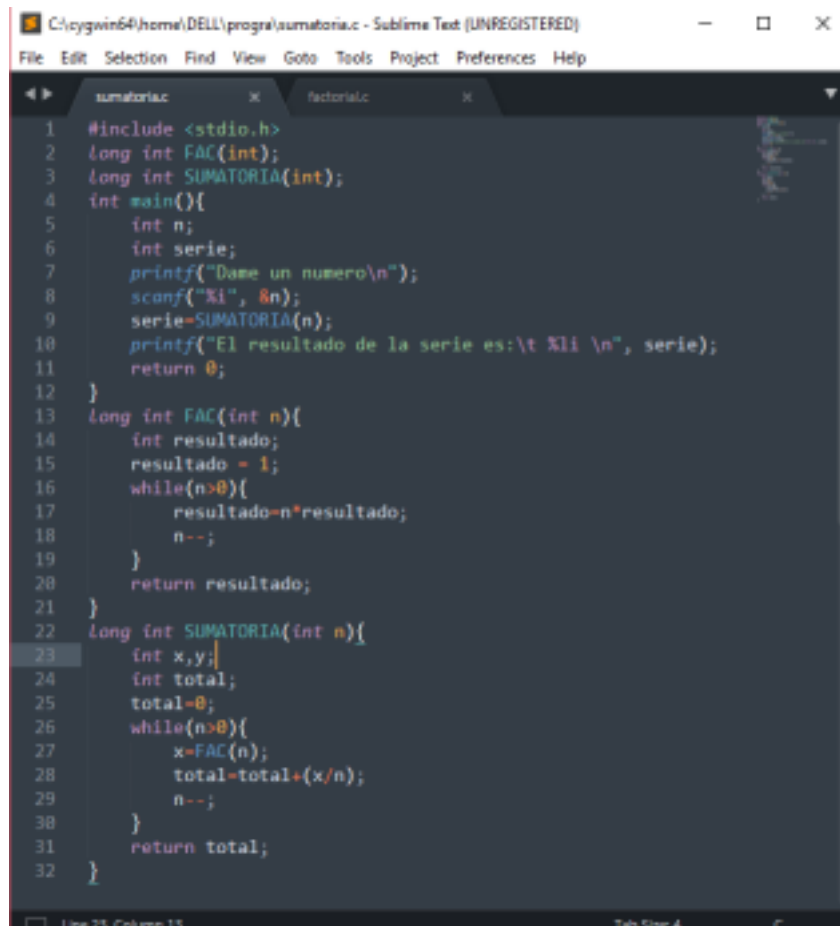
DELL@DESKTOP-NJV5QTU ~/progra
$ |
```

Al hacer la compilación podemos observar que el resultado que imprime es la factorial del numero ingresado por el usuario.

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

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

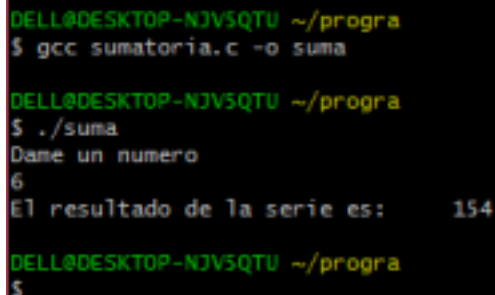
Para un numero n de entrada. Utilizar la función de factorial de la primera actividad.



```
C:\cygwin64\home\DELL\progra\sumatoria.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

sumatoria.c
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 %i \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 }
```

En este programa se utiliza la función de la factorial que anteriormente se había creado y se manda a llamar en otra nueva función, el cual devuelve como resultado la sumatoria de la serie que se solicita en esta actividad. De igual forma se hace primero los prototipos, después se mandan a llamar en la función principal y finalmente se estructura de acuerdo con lo que necesitamos.



```
DELL@DESKTOP-NJVSQTU ~/progra
$ gcc sumatoria.c -o suma

DELL@DESKTOP-NJVSQTU ~/progra
$ ./suma
Dane un numero
6
El resultado de la serie es: 154

DELL@DESKTOP-NJVSQTU ~/progra
$
```

Al hacer la compilación lo que imprime es el resultado de la sumatoria de la serie que requería esta actividad.

Conclusión:

En esta práctica se pudo apreciar que se puede colocar el nombre de las funciones y el tipo de parámetros que va a utilizar, y después elaborarlas en la parte de abajo del programa principal. Lo cual resulta conveniente pues esto permite que primero se pueda colocar las funciones que vamos a necesitar sin que anteriormente las hayamos estructurado, y de esta forma podamos ir creando las funciones de acuerdo con lo que nuestro programa vaya requiriendo. Además, de que hace el programa se vea visualmente bien, dado que podemos contemplar un menú de las funciones que ocupa y en seguida el programa principal.