

	Carátula para entrega de prácticas	
Facultad de Ingeniería	Laboratorio de docencia	

Laboratorios de computación salas A y B

<i>Profesor:</i>	Alejandro Pimentel
<i>Asignatura:</i>	Fundamentos de programación
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	12
<i>Integrante(s):</i>	Rodríguez Guzmán Paola Villanueva Bustamante Victoria

<i>No. de Equipo de cómputo empleado:</i>	44 54
<i>No. de Lista o Brigada:</i>	4926 8043 34
<i>Semestre:</i>	2020-1
<i>Fecha de entrega:</i>	3 de noviembre de 2019
<i>Observaciones:</i>	Esta carátula es inaceptable. No cumplen con el objetivo de usar prototipos de funciones.
	Además, no cumplieron con la segunda actividad, el problema no consistía solo en dividir el factorial entre un número. Cada elemento natural hasta "n" se tenía que sacar factorial y sumarlos todos. Y la implementación era en una función aparte.

CALIFICACIÓN:

5

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.

Introducción:

En lenguaje C la función principal se llama main. Cuando se ordena la ejecución del programa, se inicia con la ejecución de las instrucciones que se encuentran dentro de la función main, y ésta puede llamar a ejecutar otras funciones, que a su vez éstas pueden llamar a ejecutar a otras funciones, y así sucesivamente.

El nombre de la función se refiere al identificador con el cual se ejecutará la función; se debe seguir la notación de camello.

Actividades:

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

```
1  #include<stdio.h>
2
3  int factorial(int numero){
4      if(numero==1){
5          return 1;
6      }
7      else {
8          int a;
9          //porque si le restamos 1, seria como factorial de 5!= 4!*5
10         //ya se hace la recursividad con el factorial
11         a=numero * factorial(numero-1);
12         return a;
13     }
14 }
15 int main (){
16     int numero;
17     int resultado;
18     printf("ingrese el factorial: ");
19     scanf("%i", &numero);
20
21     resultado= factorial (numero);
22
23     printf("El factorial es: %i\n", resultado);
24     return 0;
25 }
```

No usaron prototipo
para la función

```
Documents — -bash — 80x24
[Mexico20:Documents fp03alu44$ gcc act1.c -o act
[Mexico20:Documents fp03alu44$ ./act
Ingrese un número:
4
El factorial de 1 numero es 24.
[Mexico20:Documents fp03alu44$ gcc act1.c -o act
[Mexico20:Documents fp03alu44$ ./act
Ingrese un número:
4
El factorial de numero es: 24.
[Mexico20:Documents fp03alu44$ ./act
Ingrese un número:
2
El factorial de numero es: 2.
[Mexico20:Documents fp03alu44$ ]
```

- 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.

El programa de la primera actividad se utilizó y solo se le agregó la división, entre el resultado del factorial y el número que se ingresó para calcularlo.

```

1  #include<stdio.h>
2
3  int factorial(int numero){
4      if(numero==1){
5          return 1;
6      }
7      else {
8          int a;
9          //porque si le restamos 1, seria como factorial de 5!= 4!*5
10         //ya se hace la recursividad con el factorial
11         a=numero * factorial(numero-1);
12         return a;
13     }
14 }
15 int main (){
16     int numero;
17     int resultado;
18     //se declara el resultado de la división aunque también se puede hacer direc
19     int res;
20     printf("ingrese el factorial: ");
21     scanf("%i", &numero);
22
23     resultado= factorial (numero);
24     //la división entre el numero que ingrese y ya.
25     res=resultado/numero;
26
27     printf("El factorial es: %i\n", res);
28     return 0;
29 }
30

```

Así no funcionan las sumatorias. El resultado final es la suma de cada uno de los resultados parciales divididos necesitaban una iteración extra que llamara a la función. Además eso también se tenía que hacer en una función aparte.

```

C:\Users\ANAHI\Desktop>factorial 3
ingrese el factorial: 3
El factorial es: 2

C:\Users\ANAHI\Desktop>factorial
ingrese el factorial: 5
El factorial es: 24

C:\Users\ANAHI\Desktop>factorial
ingrese el factorial: 4
El factorial es: 6

C:\Users\ANAHI\Desktop>factorial
ingrese el factorial: 6
El factorial es: 120

C:\Users\ANAHI\Desktop>

```

conclusión:

El hacer retornos de funciones es muy útil para facilitar la depuración, la mejora y el entendimiento del código, es más organizado. Fue interesante esta práctica, ya que aprendí mejor sobre el tema y nos llevó a investigar más sobre él y comprenderlo.