	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:	Alejandro E. Pimentel Alarcón
Asignatura:	Fundamentos de Programación
Grupo:	3; Bloque: 135;
No de Práctica(s):	12
Integrante(s):	Bazaldúa Morales Vanessa Torres Alcántara Alan Eliezer
No. de Equipo de cómputo	11-Macedonia
No. de Lista o	#6 ,8166; #48, 6824
Semestre:	2020-1
Fecha de entrega:	*05/Nov*/19
Observaciones:	

CALIFICACIÓN:_____

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.

Introducción:

Hasta ahora lo que habíamos hecho al realizar un programa, era ver cual era el problema y encontrar una forma de resolverlo en una sola parte, sin embargo, ahora lo visualizaremos en dos partes, para que sea más rápido encontrar solución al problema con el programa adecuado. Primero resolveremos una parte, que suponemos es la más difícil y al último solo adherimos el resto, sin complicarnos demasiado.
Además también trabajaremos con la firma, que sirve para decir a nuestro programa que existe esa función y que en algún momento la vamos a ocupar aunque todavía no este declarada, solo para que sepa que ya existe y guarde un espacio para cuando la declaremos y ocupemos.

Funciones:

Para ayudarnos a realizar las actividades que vienen a continuación, tenemos la siguiente función.

```
valorRetorno nombre (parámetros){
        // bloque de código de la función
}
```

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 el factorial de un número de entrada.

p12.c

```c
#include<stdio.h>
int a;
unsigned long long int resultado;
unsigned long long int factorial(int a){
    resultado=a;
    for(int b=1; b<a; b++){
        resultado=resultado*b;
    }
return resultado;
}
int main(){
printf("Escriba el numero factorial a calcular: ");
scanf("%i",&a);
printf("Su factorial es : %llu",factorial(a));
return 0;
}
```

En este caso declaramos primero las variables antes de la función y es válido. Además utilizamos una variables de mayor rango para que tenga una mayor capacidad de recibir números enteros positivos y podemos usar solo int como en ocasiones anteriores y también funcionará el programa pero tendrá menor rango en el resultado.
Después declaramos la función de factorial y dentro del paréntesis escribimos el valor de entrada para la función, este valor esta delimitado por los paréntesis, debemos especificar la variable que vamos a usar y además darle un nombre, como en este caso fue “a”.
Luego utilizamos un *for* que tiene como función sobreescribir “resultado” al multiplicarse con “b”, este último es un contador que va aumentando de uno en uno y se realizará la misma función hasta haber multiplicado todos los números menores al del valor de entrada “a”.

A partir de i*nt main* comienza nuestro programa principal. Recordemos que a una función se le llama al escribir su nombre y un paréntesis, que será su valor de entrada. En este caso solo tendremos un valor de entrada y este sera “n”.
Una vez realizado el programa, lo ejecutamos en la terminal y éstos fueron nuestros resultados.

```
vanessa@Titan:~/Escritorio$ gcc p12int.c -o a1
vanessa@Titan:~/Escritorio$ ./a1
Escriba el numero factorial a calcular: 3
Su factorial es : 6
vanessa@Titan:~/Escritorio$ ./a1
Escriba el numero factorial a calcular: 6
Su factorial es : 720
```

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

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

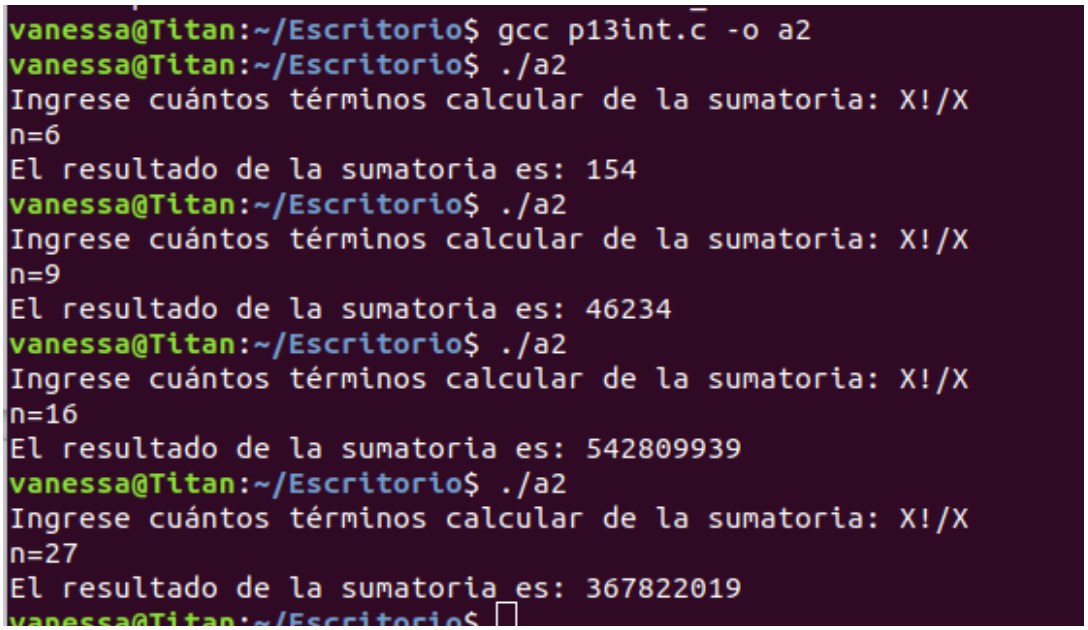
Para un número *n* de entrada. Utilizar la función de factorial de la primera actividad.

p12.c p13.c

```c
#include <stdio.h>
#include <math.h>
int a;
unsigned long long int resultado;
unsigned long long int factorial(int a){
    resultado=a;
    for(int b=1; b<a; b++){
        resultado=resultado*b;
    }
return resultado;
}
int main()
{
    unsigned long long int suma;
    int n;
    printf("Ingrese cuántos términos calcular de la sumatoria: X!/X\n");
    printf("n=");
    scanf("%i",&n);
 suma=factorial(1)/1;
    for (a=2;a<=n;a++){
        suma=suma+ factorial(a)/a;
    }
    printf("El resultado de la sumatoria es: %llu",suma);
    return 0;
}
```

Para este programa iniciamos declarando la función *factorial* de la actividad anterior, así como lo piden las indicaciones, además de que es necesario para que nuestro programa funcione.
Luego, al igual que en el anterior iniciamos el programa principal en *int main.* Después le asignamos un valor a suma, pero igual después reescribiremos su valor en el ciclo de *for.* A continuación utilizamos *for* para que se sumen todos los resultados de la serie, donde el contador *“a”* crecerá hasta que sea igual a *“n”* y todas las sumas se acumularan en un resultado final cuando el contador “a” sea igual a *“n”*. Ya que el programa realice todo este proceso al final mostrara el resultado y para esto usamos *%llu* que es el especificador para imprimir el formato de unsigned long long int, así termina el programa.

Para confirmar que el programa funcionaba, lo ejecutamos en la terminal y los resultados se muestran a continuación:



Conclusión:

Para concluir podemos decir que no es necesario declarar una función desde el principio y ahí la importancia de la firma de las funciones, no es necesario decir desde el principio lo que van a hacer simple y sencillamente debemos avisar al programas de la existencia de esa función que en algún momento vas a necesitar y después la puedes declarar, siempre y cuando este dentro del programa, éste lo va a buscar cuando la llames.