



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor : Alejandro Pimentel

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): 12

Integrante(s): Meraz Dionicio Israel

*No. de Equipo de
cómputo empleado:* 3

No. de Lista o Brigada: 8875

Semestre: 1

Fecha de entrega: 4 de Noviembre de 2019

Observaciones: Muy bien

CALIFICACIÓN: 10

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.

INTRODUCCION

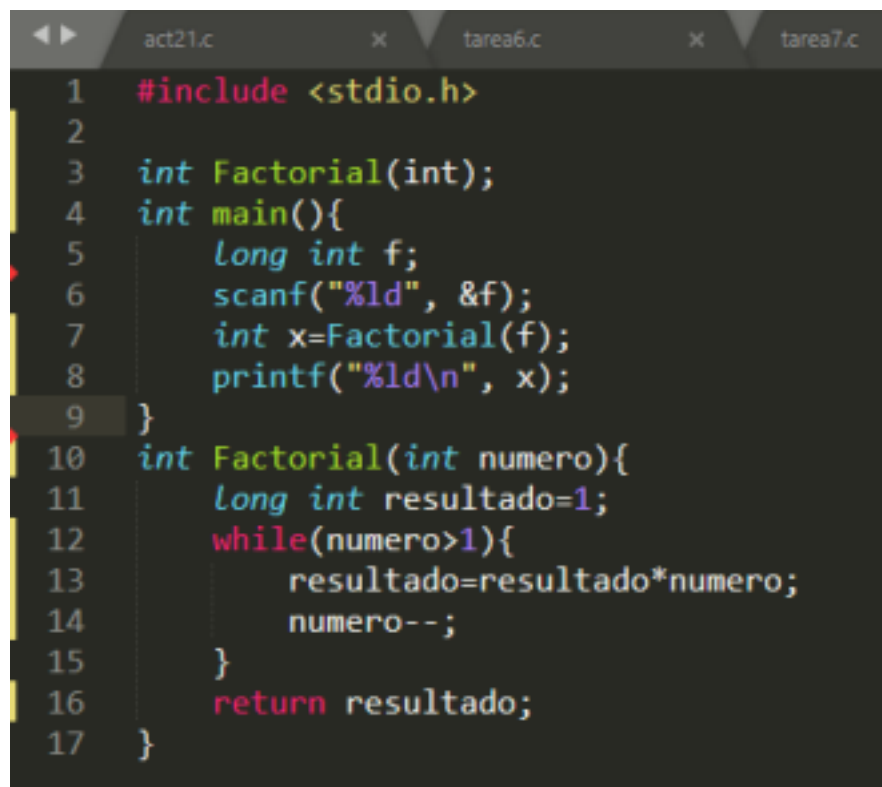
Las funciones, también llamados métodos, nos permiten dividir el trabajo que hace un programa, en tareas más pequeñas separadas de la parte principal.

Para definir una función, se utiliza la siguiente sintaxis:

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

Donde valorRetorno representa el tipo de datos del valor devuelto por la función, el nombre vaya la redundancia es el nombre de la función también se le conoce como identificador, si quieres nombrar una función no lo hagas con dos palabras, lo nombres siempre va junto y los parámetros o argumentos son variables (con especificación de tipo) donde se guardan los parámetros de la función.

- Crear un programa que regrese el factorial de un número de entrada

A screenshot of a code editor showing a C program. The editor has three tabs: 'act21.c', 'tarea6.c', and 'tarea7.c'. The code is as follows:

```
1  #include <stdio.h>  
2  
3  int Factorial(int);  
4  int main(){  
5      long int f;  
6      scanf("%ld", &f);  
7      int x=Factorial(f);  
8      printf("%ld\n", x);  
9  }  
10 int Factorial(int numero){  
11     long int resultado=1;  
12     while(numero>1){  
13         resultado=resultado*numero;  
14         numero--;  
15     }  
16     return resultado;  
17 }
```

Para esta actividad se utilizo el prototipo de función.

Primero se declaro la función Factorial y enseguida declarar la función principal y hacer el programa como si ya tuvieras la función Factorial.

Después declaramos otra vez la función pero sólo quiero que me regrese el resultado del factorial, en el return se agrega el resultado de la operación factorial.

```
ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ gcc func.c -o act

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$ ./act
5
120

ACER@EQUIPO /cygdrive/c/Users/ACER/Documents
$
```

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

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

```
1  #include <stdio.h>
2  int Factorial (int);
3  void Suma(long int);
4  int main(){
5      long int f;
6      printf("Ingresa n veces para la suma del factorial: \n");
7      scanf("%ld", &f);
8      Suma(f);
9      return 0;
10 }
11 int Factorial(int numero){
12     long int resultado=1;
13     for(int c=1; c<=numero; c++){
14         resultado=resultado*c;
15     }return resultado;
16 }
17 void Suma(long int numero){
18     long int m;
19     long int x=0;
20     for(int c=1; c<=numero; c++){
21         m= (Factorial(c))/c;
22         x=m+x;}
23     printf("El resultado es: %ld\n", x);
24 }
```

Utilizando el programa anterior de Factorial, este problema se declara las funciones que son el Factorial y Suma.

En la función Suma, otra vez se declara, lo que hice es declarar dos variables, una que guarde el factorial sucesivo que empieza de 1 hasta numero, y la otra variable para la suma de los factoriales.

CONCLUSION

Una función te puede ayudar a visualizar mejor un problema y tener en orden tu programa, en vez de tener todo las operaciones en la función principal y así perderte es mejor estar separando cada problema en funciones.

```
alumno@pcf:~/Documentos$ gcc funcion.c -o main
alumno@pcf:~/Documentos$ ./main
Ingresa n veces para la suma del factorial:
34
El resultado es: 214482817
alumno@pcf:~/Documentos$ gcc funcion.c -o main
alumno@pcf:~/Documentos$ ./main
Ingresa n veces para la suma del factorial:
2
El resultado es: 2
alumno@pcf:~/Documentos$ gcc funcion.c -o main
alumno@pcf:~/Documentos$ ./main
Ingresa n veces para la suma del factorial:
3
El resultado es: 4
alumno@pcf:~/Documentos$ ./main
Ingresa n veces para la suma del factorial:
4
El resultado es: 10
alumno@pcf:~/Documentos$ gcc funcion.c -o main
alumno@pcf:~/Documentos$ ./main
Ingresa n veces para la suma del factorial:
4
El resultado es: 10
alumno@pcf:~/Documentos$
```