



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: ALEJANDRO ESTEBAN PIMENTEL ALARCON

Asignatura: FUNDAMENTOS DE PROGRAMACION

Grupo: 03

No de Práctica(s): 12

Integrante(s): HINOJOSA RUIZ DIANA LAURA

*No. de Equipo de
cómputo empleado:* 22

No. de Lista o Brigada: 6740

Semestre: PRIMER SEMESTRE

Fecha de entrega: 04 NOVIEMBRE 2019

Observaciones: No se cumple el objetivo de utilizar una función independiente en la segunda actividad.
~~Al final debías tener dos funciones además de main~~
en donde una llamaba a la otra.

CALIFICACIÓN: 8

Objetivo.

Elaborar programas en C donde la solución del problema se divida en funciones. Distinguir lo que es 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 la presente práctica pondremos en práctica lo que hemos estado viendo últimamente en clase, el separar funciones y utilizarlas dentro de la función principal. Estas funciones se declaran antes que la principal, puede regresar algo o puede no regresar algo, además se usan a lo largo del programa, cuando requieras de ellas y ya van a estar en existencia.

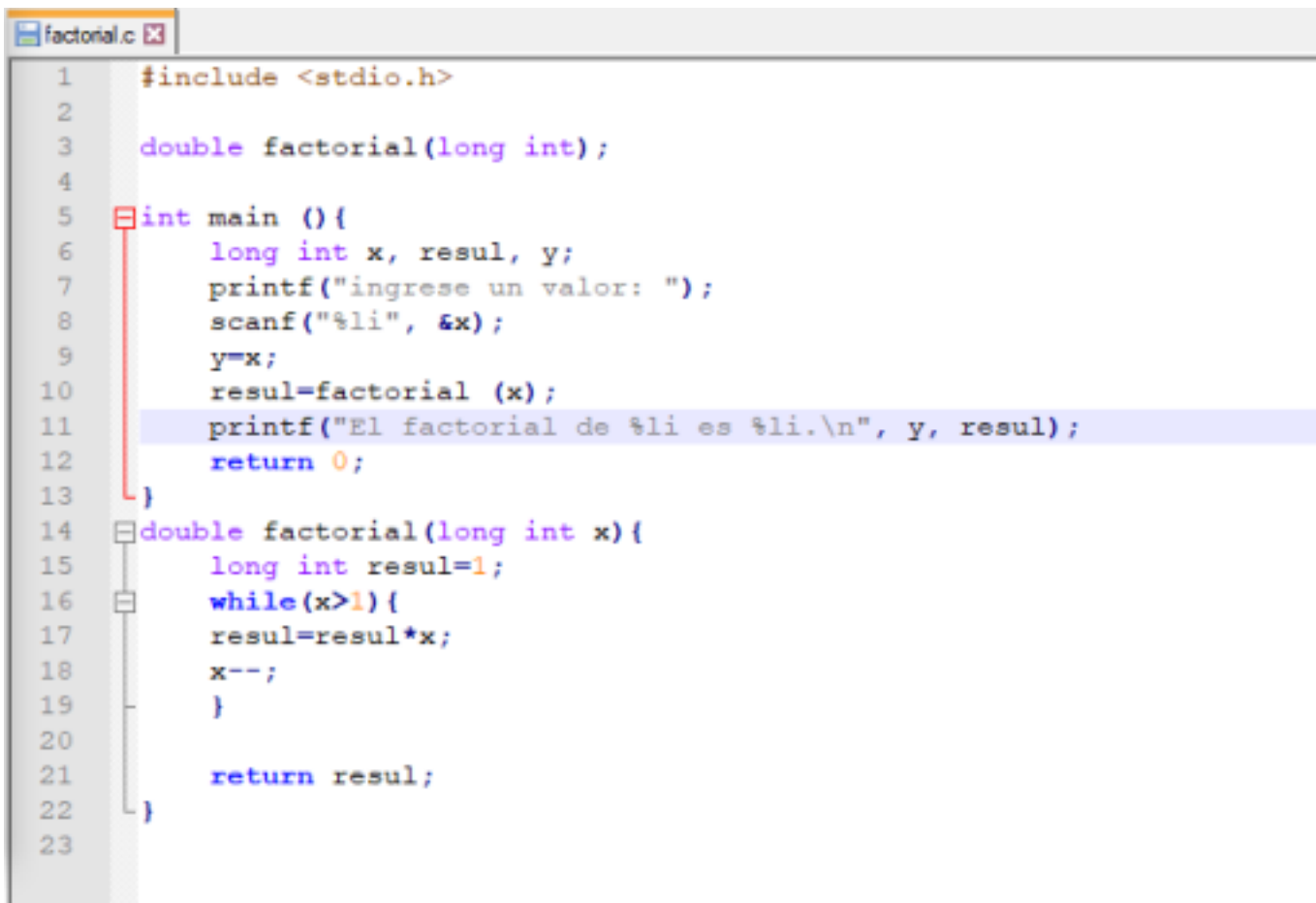
Desarrollo.

Las siguientes actividades tendrán una función antes de la principal, y la primera se va a utilizar en la segunda.

ACTIVIDAD 1:

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

Programa



```
1  #include <stdio.h>
2
3  double factorial(long int);
4
5  int main () {
6      long int x, resul, y;
7      printf("ingrese un valor: ");
8      scanf("%li", &x);
9      y=x;
10     resul=factorial (x);
11     printf("El factorial de %li es %li.\n", y, resul);
12     return 0;
13 }
14 double factorial(long int x){
15     long int resul=1;
16     while(x>1){
17         resul=resul*x;
18         x--;
19     }
20
21     return resul;
22 }
23
```

Y así es como se ve cuando corre el programa.

```
/home/programacion
factorial.c  tabla.exe  tarea3.exe  tarea6.c
primo.c     tablal.c   tarea4.c   tarea6.exe

ruus1@DESKTOP-1V77FA0 /home/programacion
$ gcc factorial.c -o factorial
factorial.c: En la función 'main':
factorial.c:11:45: error: 'resultado' undeclared (first use in this function); did you mean 'resul'?
    printf("El factorial de %li es %li.\n", y, resultado);
                                           ^~~~~~
                                           resul
factorial.c:11:45: nota: cada identificador sin declarar se reporta sólo una vez para cada función en el que aparece

ruus1@DESKTOP-1V77FA0 /home/programacion
$ gcc factorial.c -o factorial

ruus1@DESKTOP-1V77FA0 /home/programacion
$ ./factorial
ingrese un valor: 8
El factorial de 8 es 40320.

ruus1@DESKTOP-1V77FA0 /home/programacion
$ ./factorial
ingrese un valor: 4
El factorial de 4 es 24.

ruus1@DESKTOP-1V77FA0 /home/programacion
$ |
```

ACTIVIDAD 2:

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.

```
factorial.c x serie.c x serie1.c x
1  #include <stdio.h>
2
3  double factorial(long int);
4  int main(){
5      long int x, resul=0, y;
6      printf("Ingrese un valor: ");
7      scanf("%li", &x);
8      for(int i=0; i=x; i++){
9          y=factorial(x);
10         y=y/x;
11         resul=resul+y;
12         x--;
13     }
14     printf("%li\n", resul);
15     return 0;
16 }
17 double factorial(long int x){
18     long int y=1;
19     while(x>1){
20         y=y*x;
21         x--;
22     }
23     return y;
24 }
```

La sumatoria también debía ser una función aparte.

Cuando se corre el programa luce así.

```
/home/programacion
ruus1@DESKTOP-1V77FAO /home/programacion
$ gcc serie.c -o serie
ruus1@DESKTOP-1V77FAO /home/programacion
$ ./serie
9
resultado es 46234
ruus1@DESKTOP-1V77FAO /home/programacion
$ gcc serie1.c -o serie1
ruus1@DESKTOP-1V77FAO /home/programacion
$ ./serie1
Ingrese un valor: 9
46234
ruus1@DESKTOP-1V77FAO /home/programacion
$
```

Conclusión.

A demás de que se cumplió el objetivo de utilizar funciones dentro de otras funciones, también se logró hacer la segunda función utilizando las de la primera actividad, es una forma un poco más laboriosa, pero creo que te ayuda a tener un poco de más orden al hacer la función principal.