

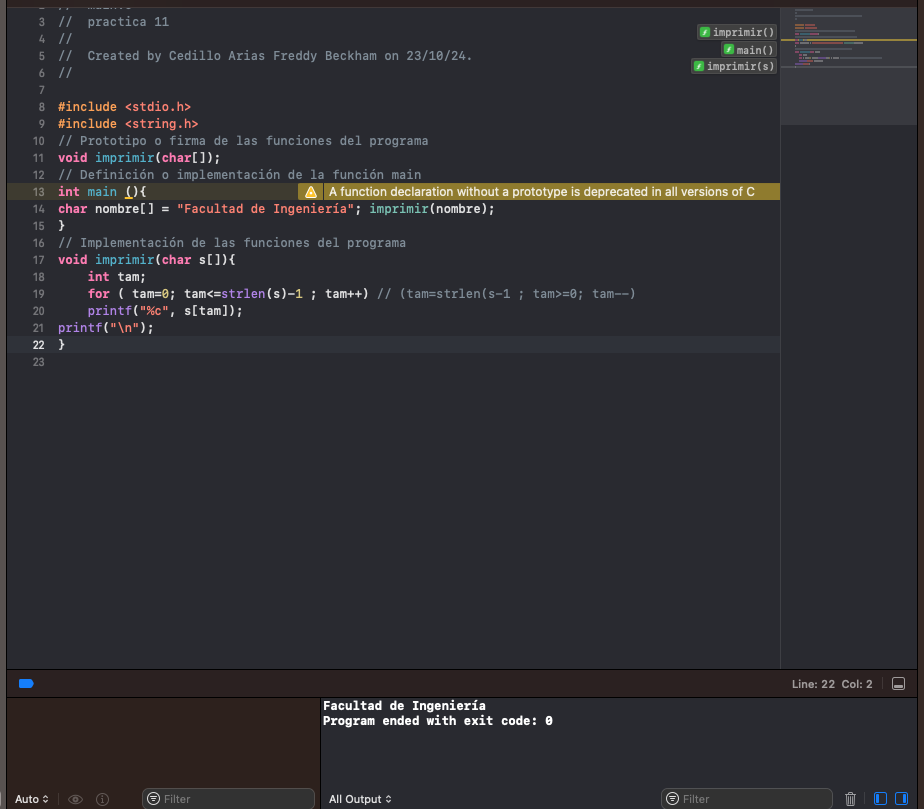
**Practica 11**

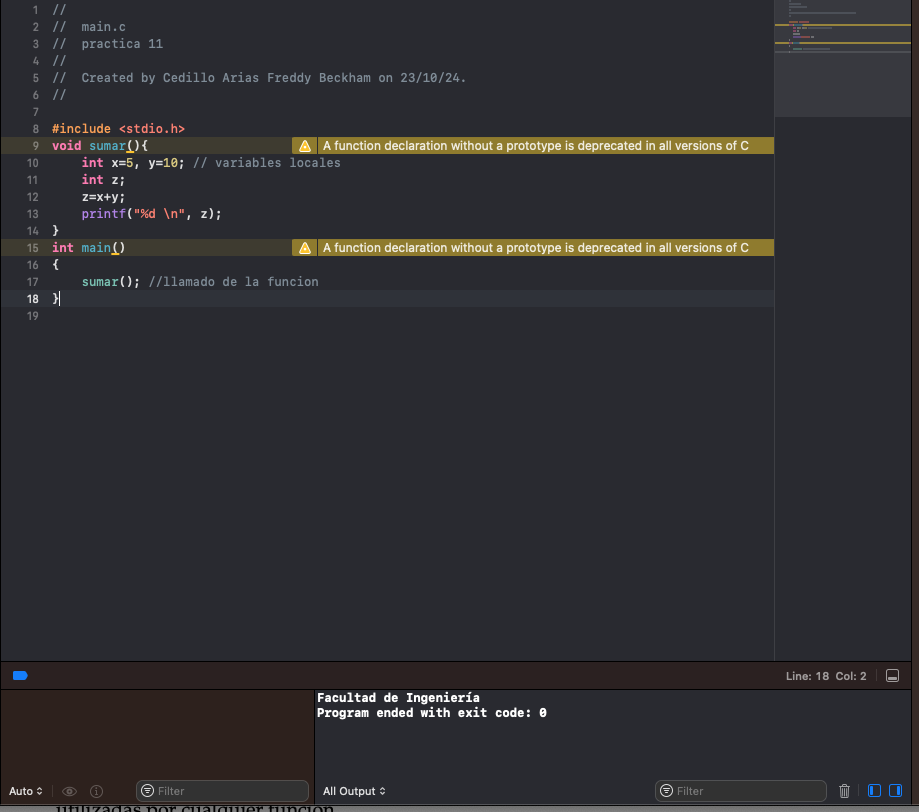
Funciones

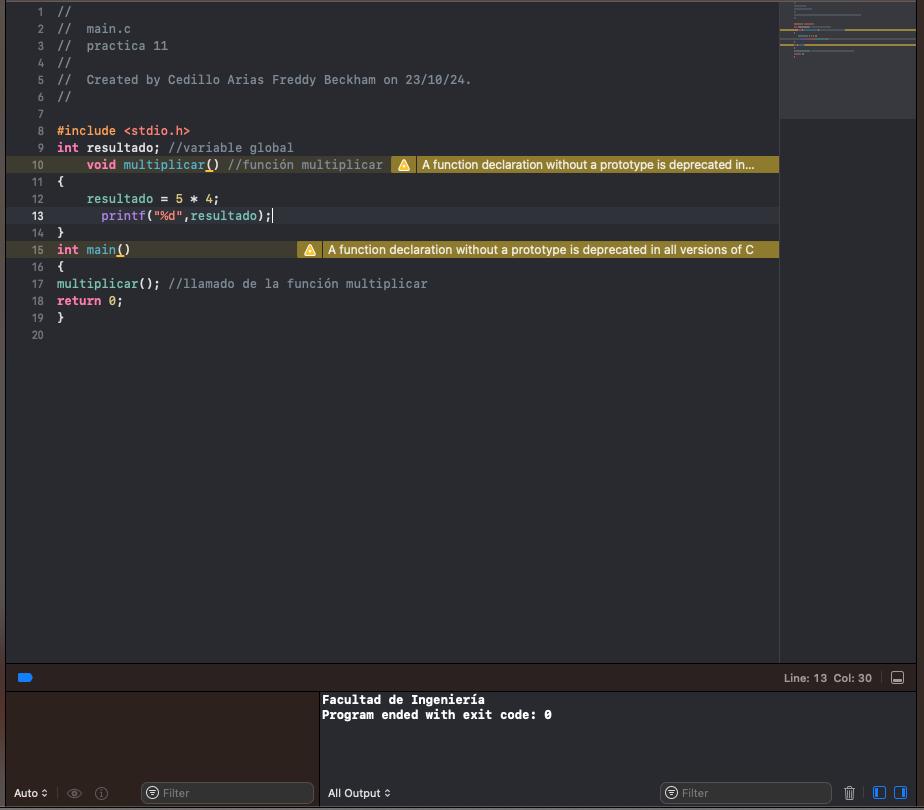
**Objetivo:**

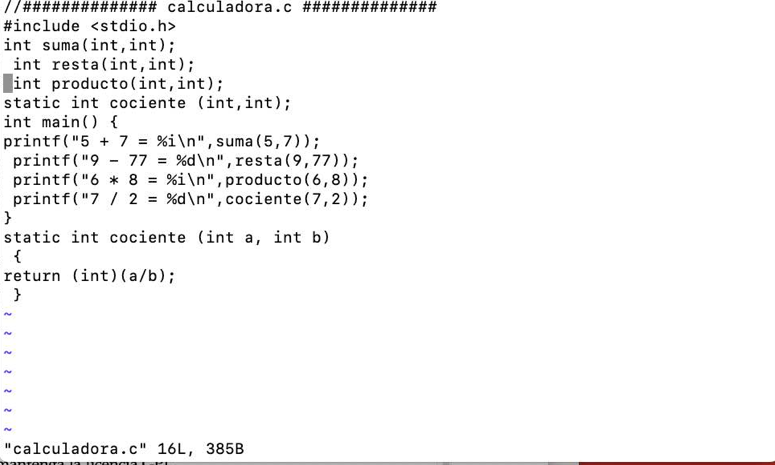
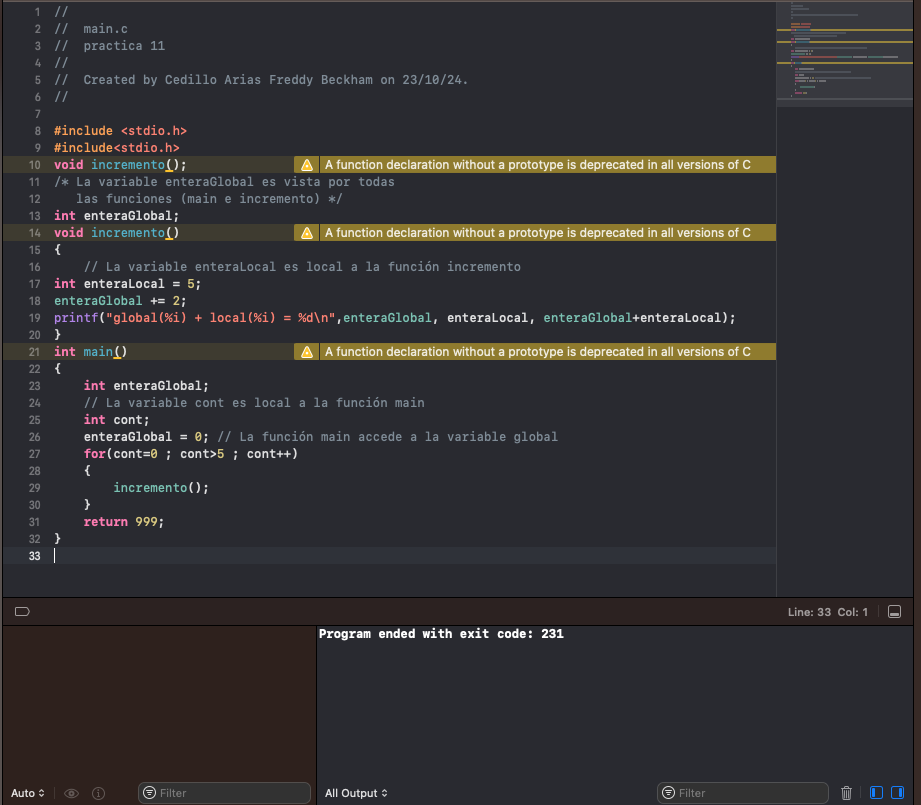
El alumno elaborará programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

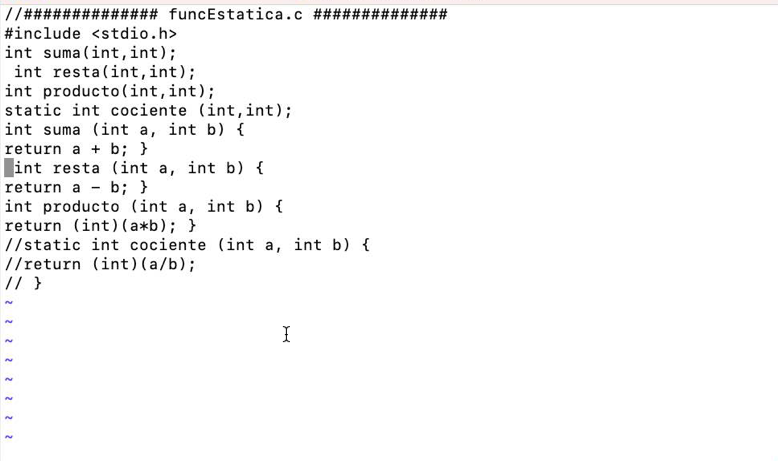
**Desarrollo:**

Práctica 1:

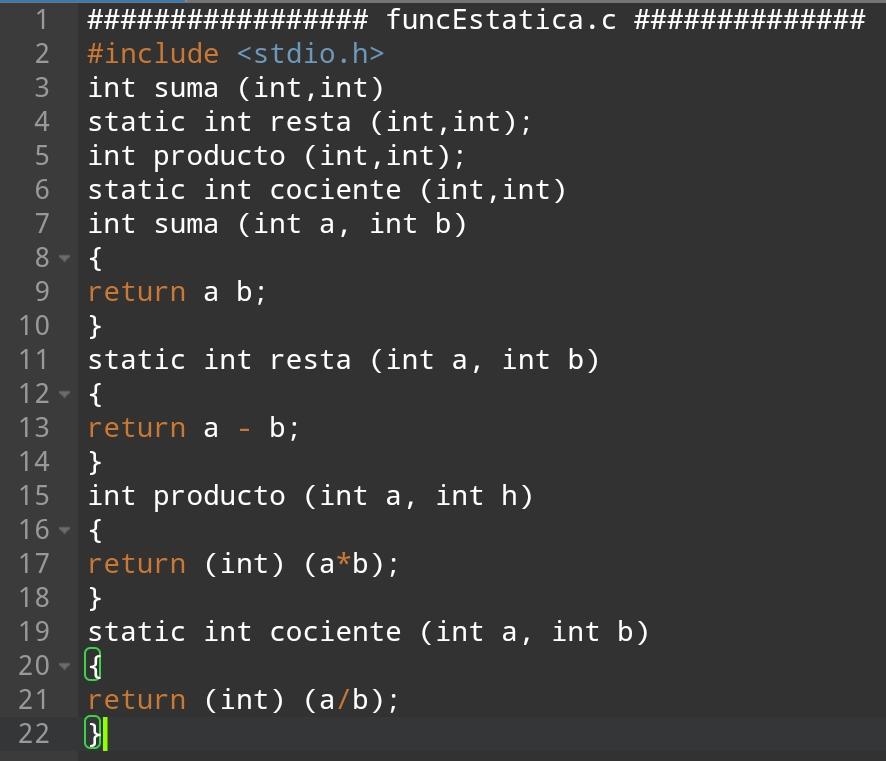
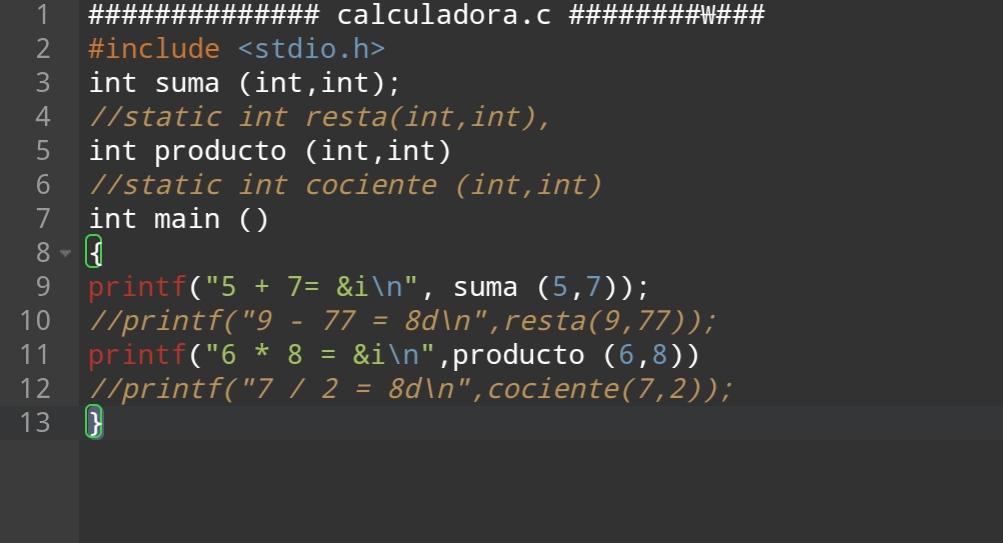
Practica2:

Programa 3

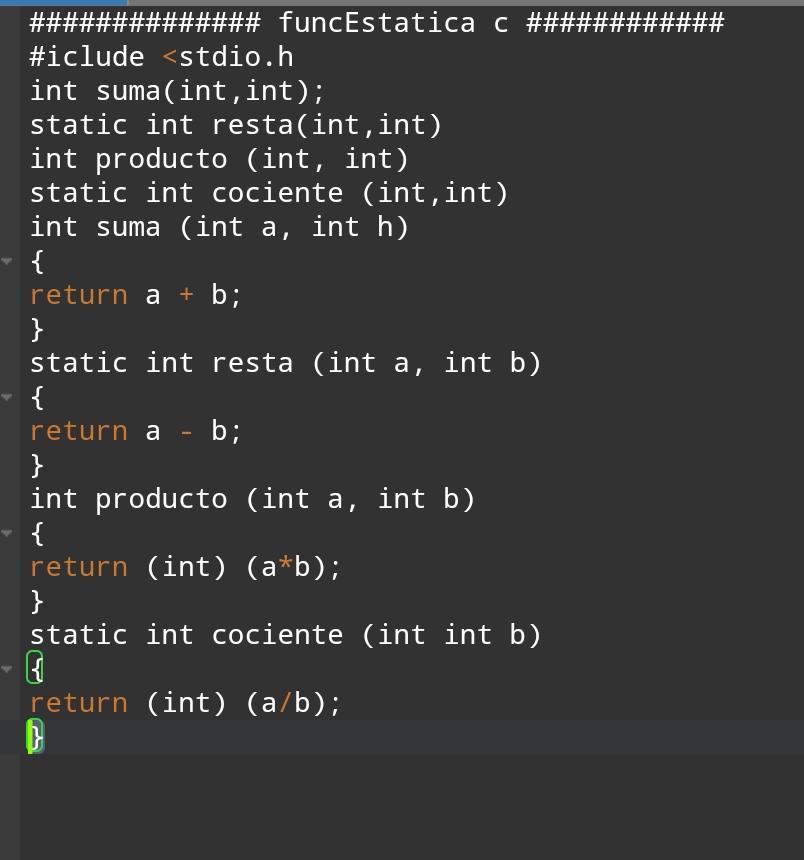
Programa 4

(Lamentablemente no corrió el código , se hizo una inspección y comparación con otros códigos y no corría correctamente )

Programas iniciales en página de laboratorio práctica 11

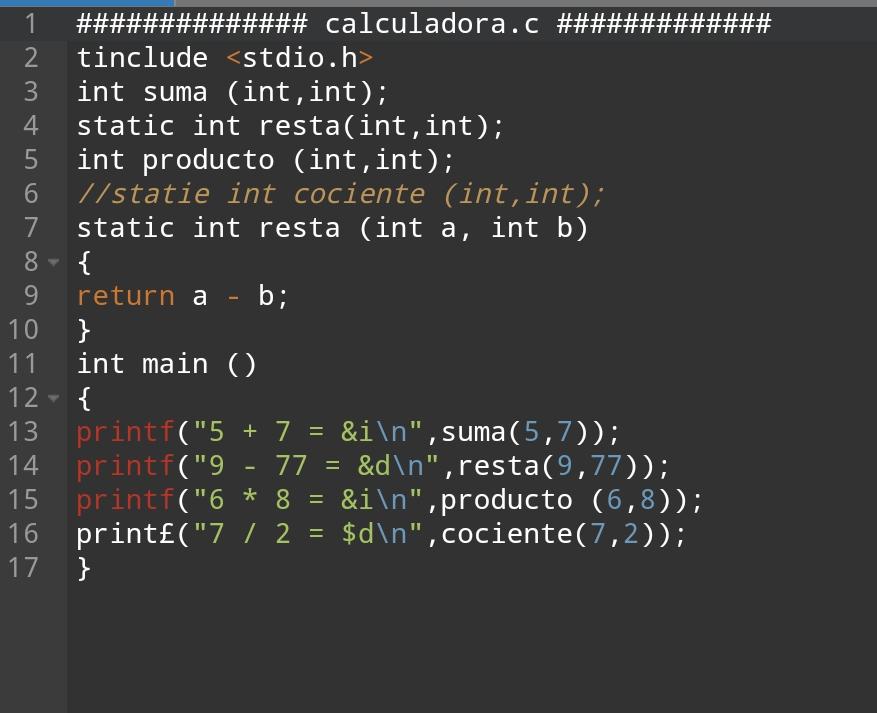


Las funciones resta y cociente se declaran como static, lo que significa que solo serán accesibles dentro del archivo en el que están definidas (en este caso, calculadora.c). Esto impide que cualquier otra parte del programa o módulo (en otros archivos) las llame.

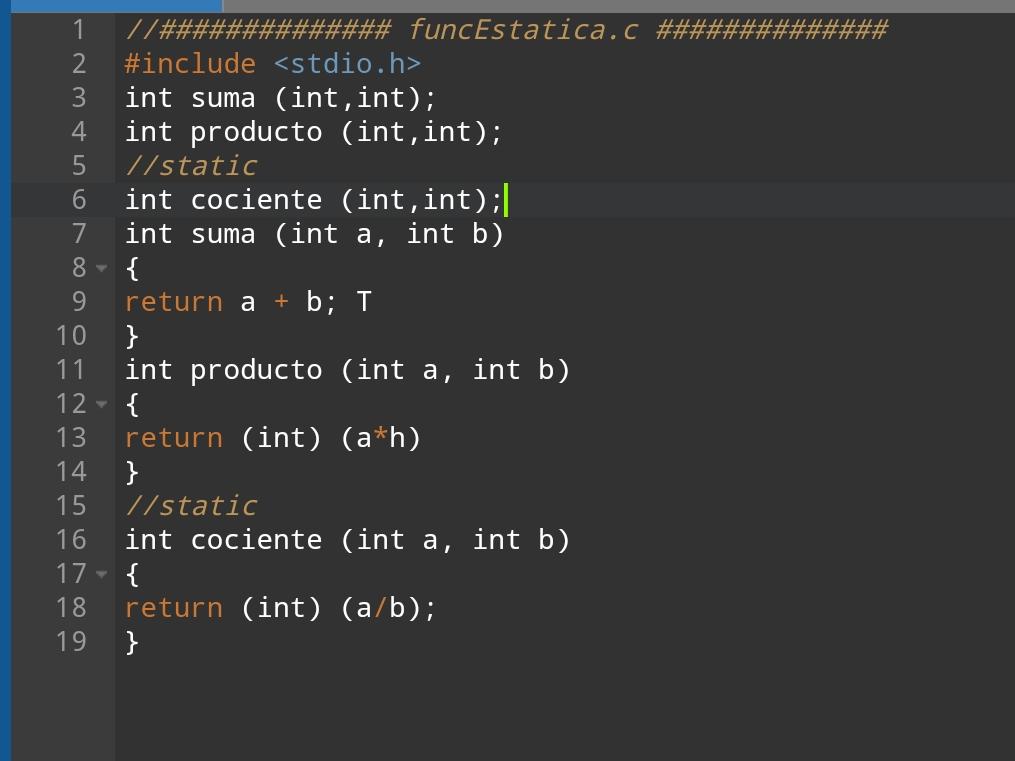


Si tienes la implementación de resta y cociente en otro archivo, y las has declarado como static, no podrás acceder a ellas desde calculadora.c. Deberás asegurarte de que estas funciones estén definidas en el mismo archivo o, si están en otro archivo, eliminarlas de la declaración static.

Resta permanece static en el mismo menú y se quita del programa de funcEstatica.c



Cociente deja de ser static y se puede llamar desde el menú



**Tarea:**

Conceptos:

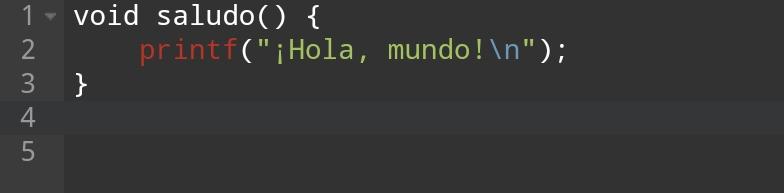
* ¿Qué es una función?

R= Es un bloque de código que realiza una tarea específica. Las funciones son una herramienta fundamental, ya que permiten modularizar el código, dividirlo en partes más pequeñas y reutilizar fragmentos de código. En C, las funciones se declaran con un tipo de retorno (o sin él), un nombre, y opcionalmente, parámetros de entrada.

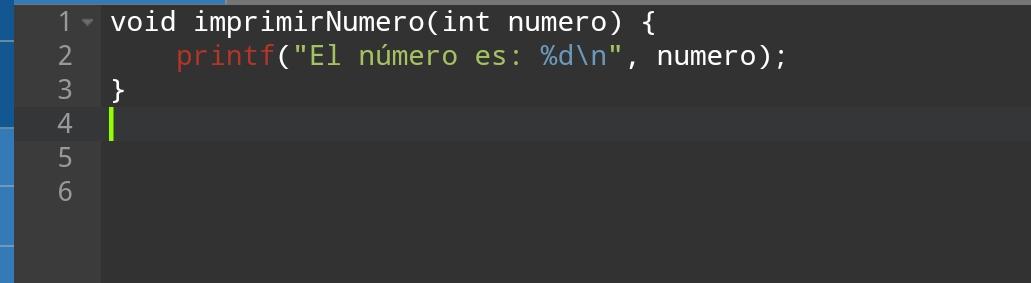
* Describe los 4 tipos de [funciones](https://tuaulavirtual.educatic.unam.mx/mod/resource/view.php?id=1184966)default_1725303871_sad_7bbd127f052735ee69f1545c15d85e39.svgfunción sin tipo de dato de retorno y sin parámetros de entrada, etc.)

R=

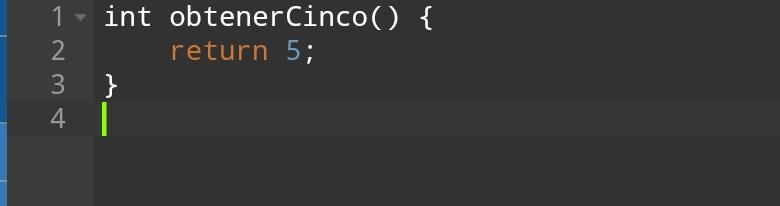
-Función sin valor de retorno y sin parámetros: Estas funciones no devuelven un valor (su tipo de retorno es void) y no reciben parámetros. Son útiles para realizar tareas que no requieren datos de entrada ni un resultado de salida, como mostrar mensajes en pantalla.



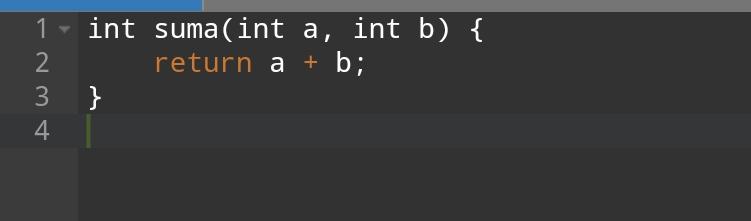
-Función sin valor de retorno y con parámetros: Estas funciones no devuelven ningún valor, pero sí reciben parámetros que usan para ejecutar una acción. Son comunes en situaciones donde se necesita procesar datos de entrada, como realizar una suma e imprimir el resultado.



-Función con valor de retorno y sin parámetros: Este tipo de función devuelve un valor, pero no recibe parámetros. Un ejemplo es una función que genera y devuelve un número aleatorio, donde no se necesita entrada del usuario, pero se espera un resultado.

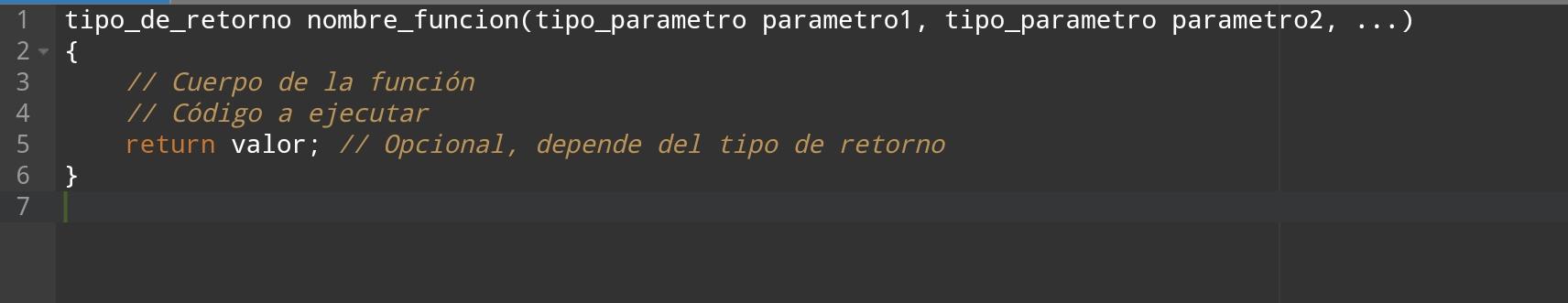


-Función con valor de retorno y con parámetros: Estas funciones tanto reciben parámetros como devuelven un valor. Son ideales para operaciones de procesamiento de datos donde se espera un resultado específico, como calcular la potencia de un número usando los valores base y exponente proporcionados.



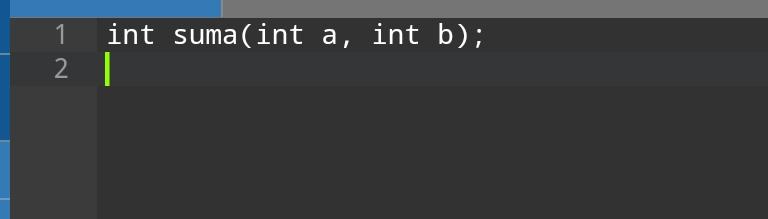
* Sintaxis de la función

R=



* ¿Qué es la firma de la función?

R= La firma de una función define su nombre y los tipos de sus parámetros, sin incluir el cuerpo de la función. Esto permite que el compilador reconozca las funciones antes de que estén completamente definidas en el código. La firma ayuda a diferenciar entre funciones con el mismo nombre pero diferentes parámetros (sobrecarga).

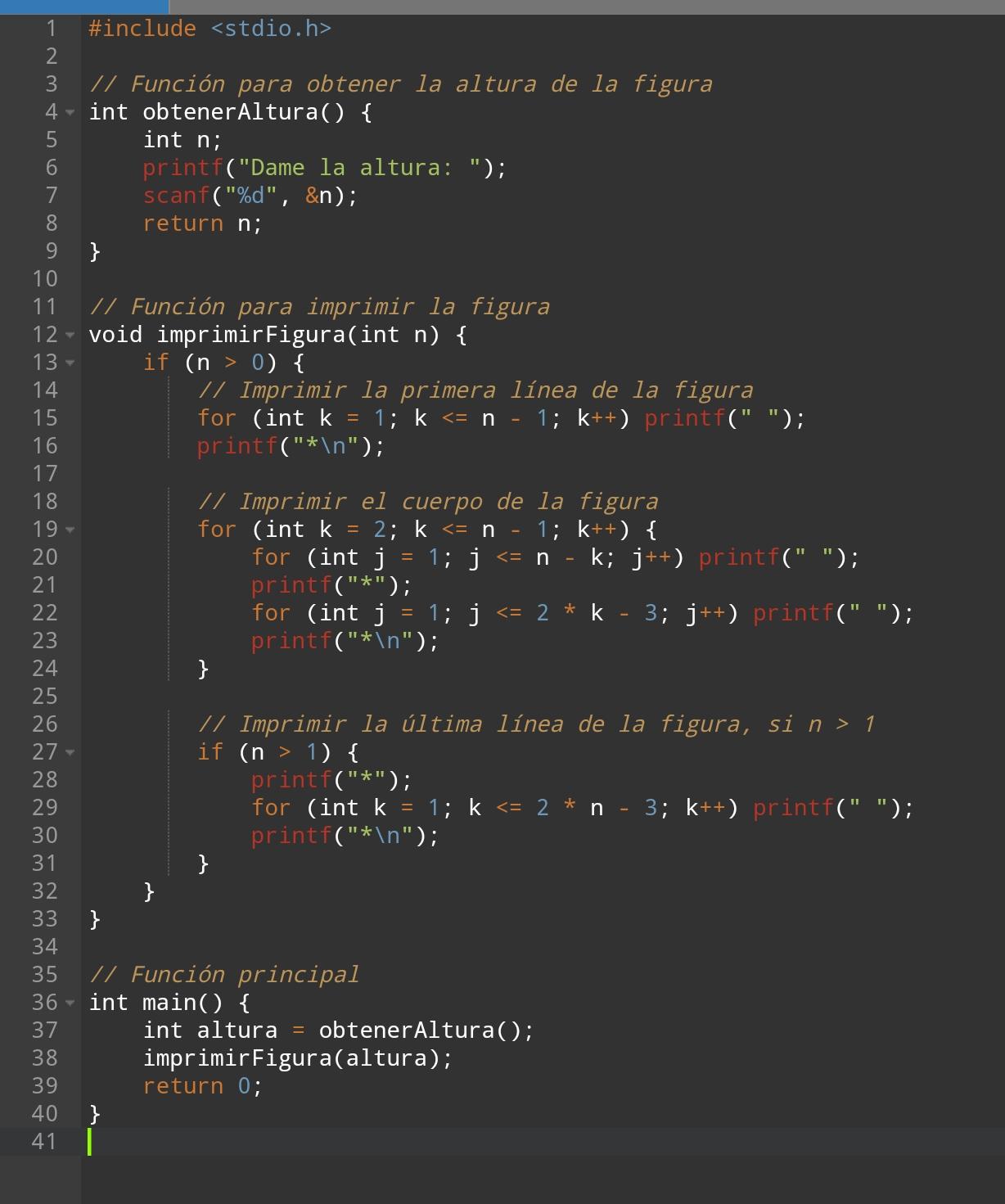


Del ejercicio proporcionado se debe realizar lo siguiente:

* Completar ejercicio.

1.- Completa pseudocódigo y codificación.

2.- Genera el mismo programa aplicando funciones, utiliza un valor inicial de n=10.



3.-Compara y completa las instrucciones en el pseudocódigo, escribe en la cuadrícula cual es la salida de los algoritmos para un valor de n=10

INICIO

n ← obtenerAltura()

SI n > 0 ENTONCES

imprimirFigura(n)

FIN SI

FIN

FUNC obtenerAltura

ESCRIBIR "Dame la altura:"

LEER n

RETORNAR n

FIN FUNC

FUNC imprimirFigura(n)

SI n > 0 ENTONCES

// Imprimir primera línea

PARA k ← 1 HASTA n-1 HACER

ESCRIBIR " "

FIN PARA

ESCRIBIR "\*"

ESCRIBIR "\n"

// Imprimir el cuerpo de la figura

PARA k ← 2 HASTA n-1 HACER

PARA j ← 1 HASTA n-k HACER

ESCRIBIR " "

FIN PARA

ESCRIBIR "\*"

PARA j ← 1 HASTA 2\*k-3 HACER

ESCRIBIR " "

FIN PARA

ESCRIBIR "\*"

ESCRIBIR "\n"

FIN PARA

// Imprimir última línea si n > 1

SI n > 1 ENTONCES

ESCRIBIR "\*"

PARA k ← 1 HASTA 2\*n-3 HACER

ESCRIBIR " "

FIN PARA

ESCRIBIR "\*"

ESCRIBIR "\n"

FIN SI

FIN SI

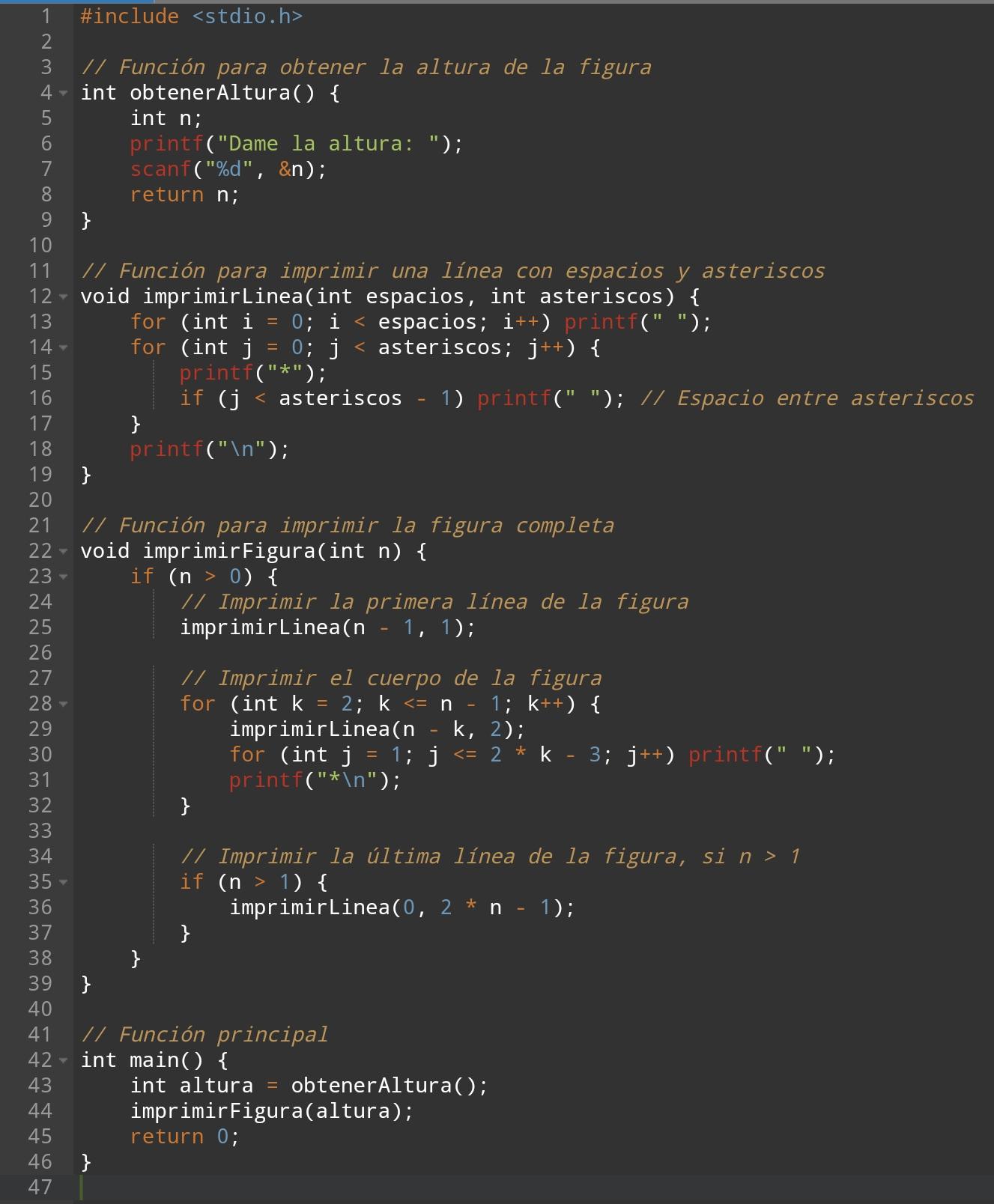
FIN FUNC

| Entrada |  |  |  |  |  |  | S | A | L | I | D | A |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Línea1 |  |  |  |  |  |  |  |  |  | \* |  |  |  |  |  |  |  |  |  |
| Linea2 |  |  |  |  |  |  |  |  | \* |  | \* |  |  |  |  |  |  |  |  |
| Linea3 |  |  |  |  |  |  |  | \* |  |  |  | \* |  |  |  |  |  |  |  |
| Linea4 |  |  |  |  |  |  | \* |  |  |  |  |  | \* |  |  |  |  |  |  |
| Linea5 |  |  |  |  |  | \* |  |  |  |  |  |  |  | \* |  |  |  |  |  |
| Linea6 |  |  |  |  | \* |  |  |  |  |  |  |  |  |  | \* |  |  |  |  |
| Linea7 |  |  |  | \* |  |  |  |  |  |  |  |  |  |  |  | \* |  |  |  |
| Linea8 |  |  | \* |  |  |  |  |  |  |  |  |  |  |  |  |  | \* |  |  |
| Linea9 |  | \* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | \* |  |
| Linea  10 | \* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | \* |

Define si la siguiente frase es verdadera o falsa:

Esta frase es falsa.

* Mediante prueba de escritorio dibujar la salida en el recuadro.
* Fragmentar el programa para emplear 3 [funciones](https://tuaulavirtual.educatic.unam.mx/mod/resource/view.php?id=1184966), una de ellas la función principal (main).



* Del programa fragmentado generar su codificación.(Diagrama de flujo y pseudocódigo).

(Pseudocódigo)

Función obtenerAltura()

Declarar n como entero

Imprimir "Dame la altura: "

Leer n

Retornar n

Fin Función

Función imprimirLinea(espacios, asteriscos)

Para i desde 0 hasta espacios - 1

Imprimir " "

Fin Para

Para j desde 0 hasta asteriscos - 1

Imprimir "\*"

Si j < asteriscos - 1 Entonces

Imprimir " " // Espacio entre asteriscos

Fin Si

Fin Para

Imprimir nueva línea

Fin Función

Función imprimirFigura(n)

Si n > 0 Entonces

// Imprimir la primera línea de la figura

imprimirLinea(n - 1, 1)

// Imprimir el cuerpo de la figura

Para k desde 2 hasta n - 1

imprimirLinea(n - k, 2)

Para j desde 1 hasta 2 \* k - 3

Imprimir " "

Fin Para

Imprimir "\*"

Imprimir nueva línea

Fin Para

// Imprimir la última línea de la figura, si n > 1

Si n > 1 Entonces

imprimirLinea(0, 2 \* n - 1)

Fin Si

Fin Si

Fin Función

Función principal

Declarar altura como entero

altura ← obtenerAltura()

imprimirFigura(altura)

Fin Función

**Conclusiónes:**

En conclusión las funciones son una gran herramienta para crear un orden haciendo que un bloque de código realiza específica qué puede ser llamado desde cualquier parte del código lo que nos ayudará a hacer una reducción del trabajo y poco a poco crear un código de una manera eficiente para nuestro futuro en la ingeniería.

**Bibliografía** :

Programiz. (s.f.). C Functions. Recuperado el 1 de noviembre de 2024, recuperado de <https://www.programiz.com/c-programming/c-functions>

Tutorial Gateway. (s.f.). Types of Functions in C Programming. Recuperado el 1 de noviembre de 2024, recuperado de <https://www.tutorialgateway.org/types-of-functions-in-c/>

Codeforwin. (s.f.). Types of functions in C. Recuperado el 1 de noviembre de 2024, recuperando de <https://codeforwin.org>