

**Título de la práctica**

Guía práctica de estudio 07: Fundamentos de Lenguaje C

**Objetivo:**

Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

**Desarrollo de la práctica**

El lenguaje de programación C fue creado por Brian Kernighan y Dennis

Ritchie a mediados de los años 70. La primera implementación del mismo la realizó

Dennis Ritchie sobre un computador DEC PDP-11 con sistema operativo UNIX. C es

el resultado de un proceso de desarrollo que comenzó con un lenguaje anterior, el

BCPL, el cual influyó en el desarrollo por parte de Ken Thompson de un lenguaje

llamado B, el cual es el antecedente directo del lenguaje C. El lenguaje C es un

lenguaje para programadores en el sentido de que proporciona una gran flexibilidad de

programación y una muy baja comprobación de incorrecciones, de forma que el

lenguaje deja bajo la responsabilidad del programador acciones que otros lenguajes

realizan por si mismos. Así, por ejemplo, C no comprueba que el índice de referencia

de un vector (llamado array en la literatura informática) no sobrepase el tamaño del

mismo; que no se escriba en zonas de memoria que no pertenecen al área de datos del

programa, etc.

El lenguaje C es un lenguaje estructurado, en el mismo sentido que lo son otros

lenguajes de programación tales como el lenguaje Pascal, el Ada o el Modula-2, pero

no es estructurado por bloques, o sea, no es posible declarar subrutinas (pequeños

trozos de programa) dentro de otras subrutinas, a diferencia de como sucede con otros

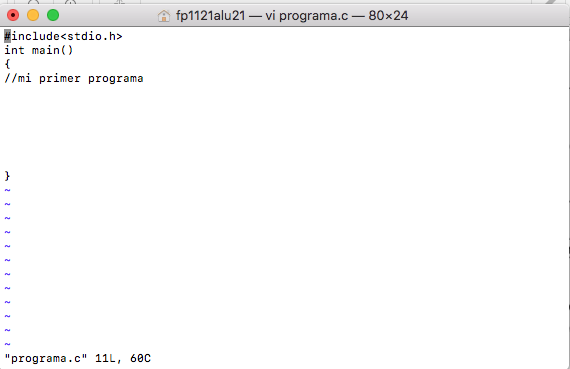
lenguajes estructurados tales como el Pascal. Además, el lenguaje C no es rígido en la

comprobación de tipos de datos, permitiendo fácilmente la conversión entre diferentes

tipos de datos y la asignación entre tipos de datos diferentes.

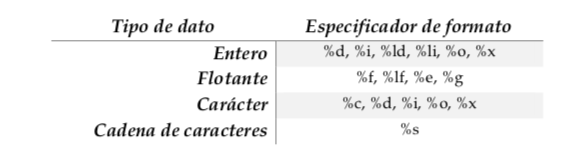
- Identificadores. Antes de proceder a explicar los identificadores en C, es necesario resaltar que C es un lenguaje sensible al contexto, a diferencia por ejemplo de Pascal, por lo cual, C diferencia entre mayúsculas y minúsculas, y por tanto, diferencia entre una palabra escrita total o parcialmente en mayúsculas y otra escrita completamente en minúsculas. En el lenguaje C, un identificador es cualquier palabra no reservada que comience por una letra o por un subrayado, pudiendo contener en su interior letras, números y subrayados. La longitud máxima de un identificador depende del compilador que se este usando, pero, generalmente, suelen ser de 32 caracteres, ignorándose todos aquellos caracteres que compongan el identificador y sobrepasen la longitud máxima. Recuérdese, además, que al ser C sensible al contexto, un identificador escrito como esto\_es\_un\_ident y otra vez como Esto\_Es\_Un\_Ident será interpretado como dos identificadores completamente distintos.

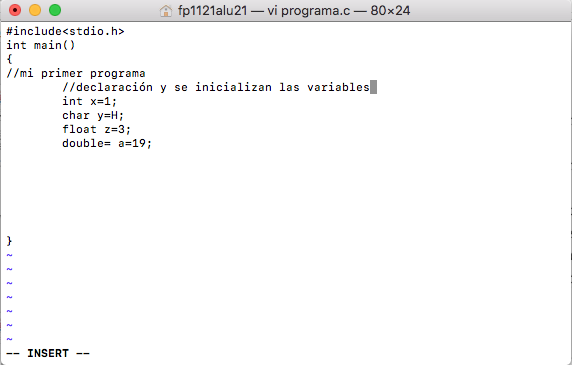
**Comentario en línea**

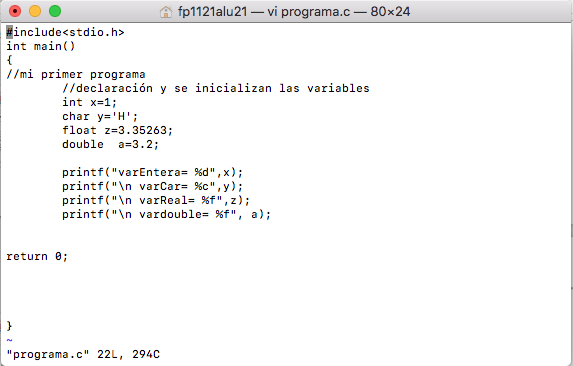
****

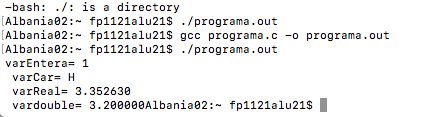
Declaración de variables

Tipos de datos (int,char,float, double)

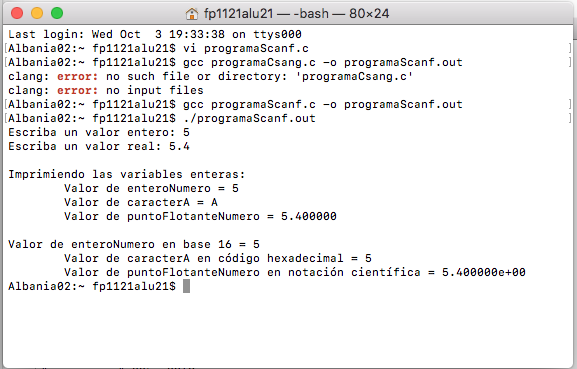
  
Inicializar variables



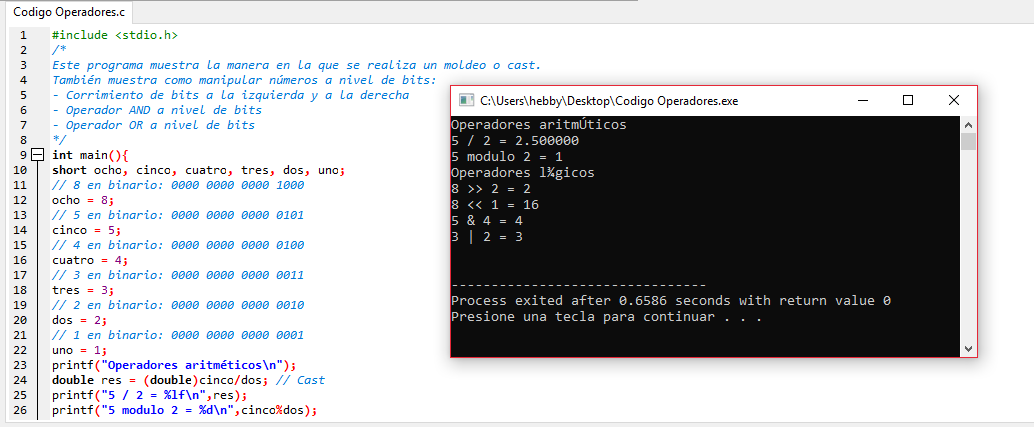




**Código almacenar e imprimir variables**

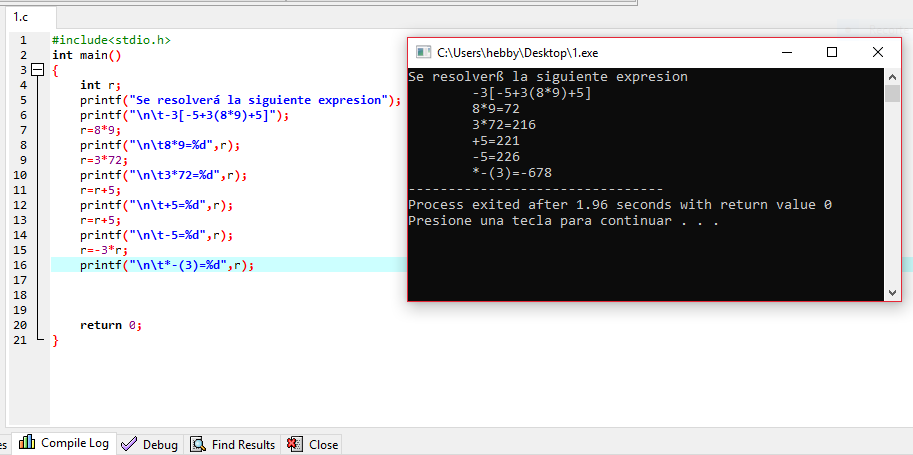


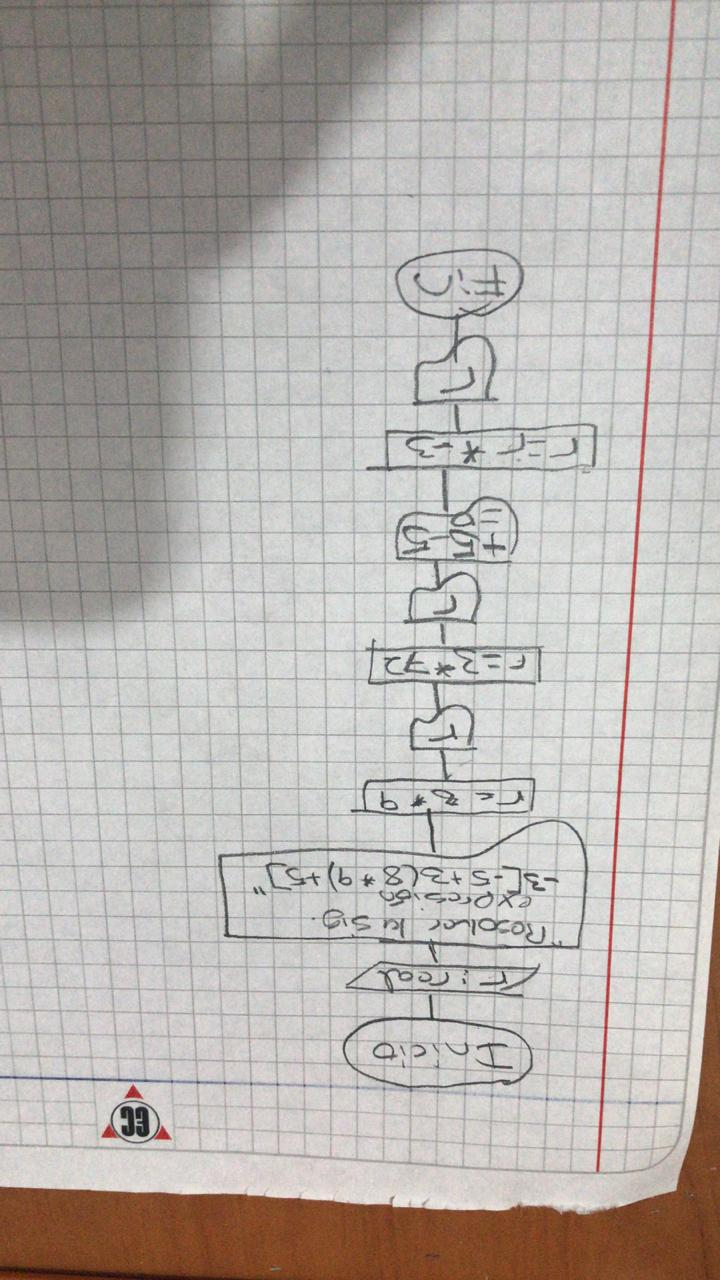
**Código operadores**

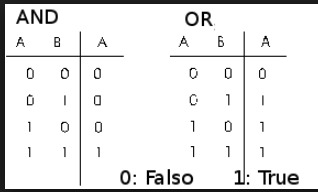


**Ejercicios de Tarea**

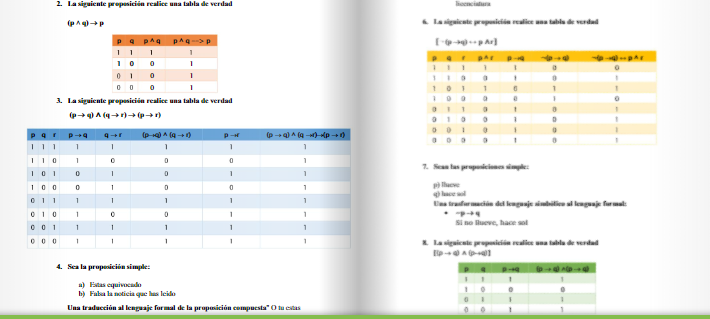
1







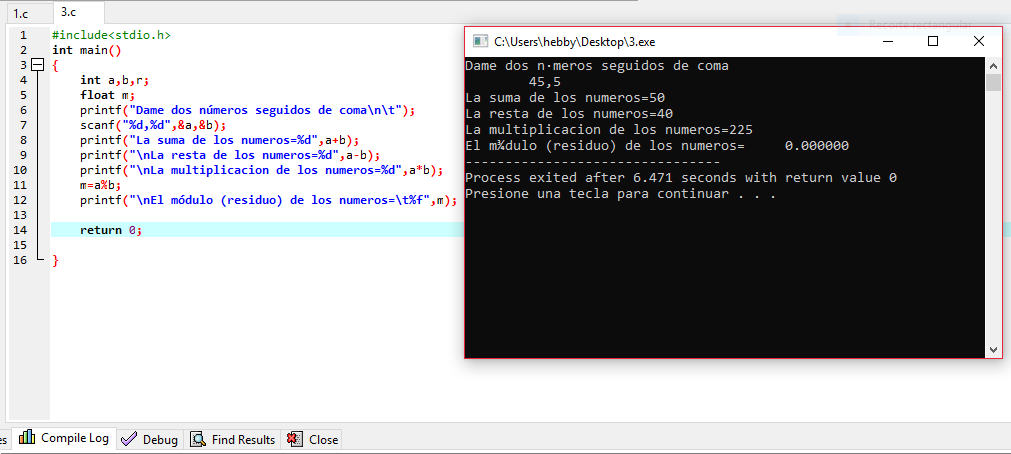
2

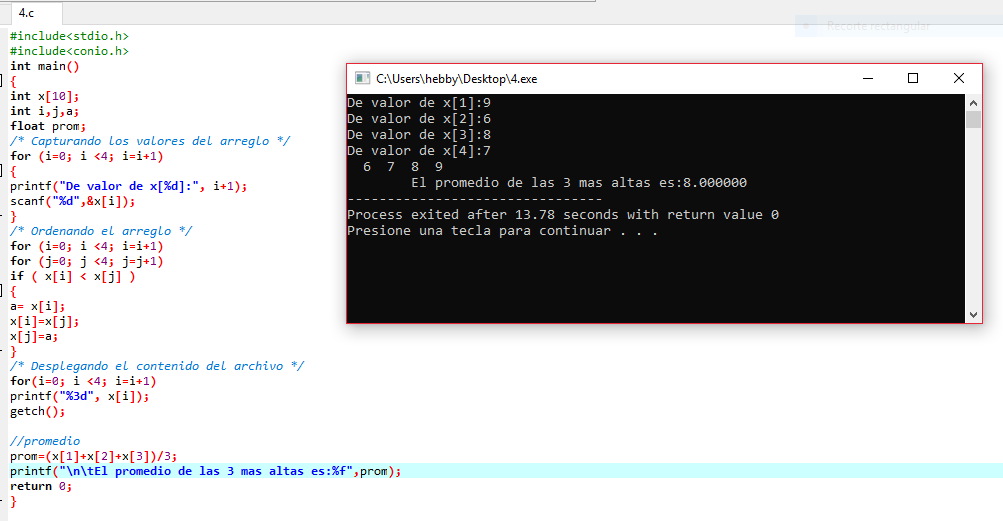


3

Análisis: Entrada: 2 números separados por coma

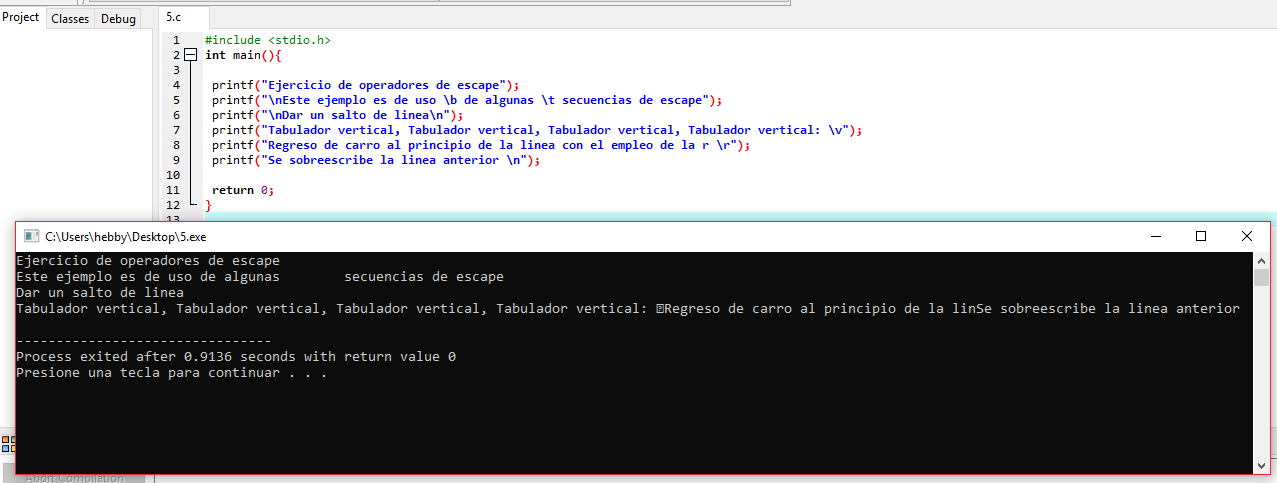
Salida: suma, resta, multiplicación y módulo





4

5



El programa 5 ejemplifica los operadores de escape como lo es \b (backspace: tu cursor se va a a imprimir sin dar un espacio entre letras) ,\n(Salto de línea),\t(Es un tabulador),\r( La "\r" significa retorno de carro, es decir, la tecla enter)|.

**Conclusiones**

Mis observaciones hacia la práctica es que estamos formando un gran conocimiento teniendo como clave las bases para poder programar, la cada práctica aumentarle la dificultad y no solo eso, agregarle un conocimiento extra nos hace estar mejor preparados para el día futuro.

El empezar a usar la terminal ya para ver el funcionamiento de los programas es base para así saber los comandos que en ella se requieren, pudiendo analizar de manera directa como se tiene que declarar e iniciar variables, el tipo de dato y sus maneras de detallarlo en la terminal.

Lo bueno de trabajar en C es que se puede trabajar el mismo programa en MAC que en PC y al tener las dos variantes podemos ir adquiriendo experiencias paralelas.

**Bibliografía**

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda  
edición, USA, Pearson Educación 1991.

Carlos Guadalupe (2013). Aseguramiento de la calidad del software (SQA). [Figura 1].