|  |
| --- |
| Laboratorio de Computación  Salas A y B |
|  |

|  |  |
| --- | --- |
| Profesor(a): | García Morales Karina |
| Asignatura: | Fundamentos de Programación |
| Grupo: | 22 |
| No de Práctica(s): | 6 |
| Integrante(s): | Viveros Fernández Salvador Juan |
|  |  |
|  |  |
|  |  |
| No. de lista o brigada: | 54 |
| Semestre: | Primero |
| Fecha de entrega: | 27/09/2024 |
| Observaciones: |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Entorno y fundamentos de lenguaje C**

**Objetivos:**

El alumno elaborará 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:**

**Ciclo de vida de Software:**

De acuerdo con el ciclo de vida del software. Al momento en el que se hayan analizado los datos de entrada y salida, se haya diseñado un algoritmo y se haya representado a este mismo en de manera gráfica con un diagrama de flujo, o de forma escrita en pseudocódigo, es momento de ir al siguiente paso, la codificación y sus posteriores pruebas, siendo usado en este caso, el lenguaje C.

**Entorno de C:**

Un lenguaje de programación permite escribir una serie de instrucciones, las cuales, tras ser procesadas por un compilador, pueden ser interpretadas y ejecutadas por una máquina.

El lenguaje C presenta la ventaja de ser un lenguaje de alto nivel, al ser de fácil de programar, al mismo tiempo que es fácil para la máquina de interpretar, siendo esta una característica de un lenguaje de bajo nivel. De esta manera, se ha convertido en uno de los programas más utilizados en la industria

**Compilador:**

Un compilador es un programa que toma como datos de entrada un archivo de texto creado por un desarrollador, dando como salida un programa ejecutable. El lenguaje C tiene como ventaja que, siempre y cuando se siga su estándar, puede ser ejecutado en cualquier máquina, siempre y cuando ésta tenga un compilador de este mismo lenguaje, siendo así, adquiere el título de multiplataforma.

Si se aleja del estándar, el programa no podrá ejecutarse en todos los dispositivos, por lo que será necesario hacer modificar el código y realizar correcciones. Por otra parte, si se comete un error en la sintaxis del lenguaje y un error puede desencadenar muchos más, por lo que el compilador no proporcionará ningún ejecutables, dejando en su lugar un código de error, teniéndose que utilizar algún editor para poder solucionarlo. No obstante, si el compilador sólo arroja alguna advertencia, éste si proporcionará un ejecutable, pero cabe la posibilidad de que el programa sufra diversos problemas a la hora de ejecutarse, siendo recomendable revisar el código y corregir los errores existentes.

**Editor Visual de Interface de GNU/Linux (vi):**

El editor *vi* (virtual interface por sus siglas en inglés) es el editor que se puede encontrar más comúnmente en cualquier sistema con núcleo basado en UNIX, encontrándose disponible en línea de comandos, teniendo como ventaja que puede llegar a ser el único disponible dependiendo del sistema.

Para iniciar el editor, creando o abriendo un archivo, se debe de escribir el comando *vi nombre\_archivo[.ext]* siendo el .ext la extensión del archivo a trabajar, siendo en este caso *.c*.

Así mismo, esta interfaz posee comandos para poder ejecutar acciones en el modo por defecto y no se puede modificar el texto libremente, siendo algunos ejemplos:

-*k* mueve el cursor hacia arriba.

-*j* mueve el cursor hacia abajo

-*h* mueve el cursor de la izquierda

-*l* mueve el cursor hacia la derecha

-*yy* copia la línea donde está el cursor

-*p* pega un contenido copiado o borrado

-*u* deshace el último cambio.

Así mismo, este posee una modalidad llamada modo comando, donde se pueden ejecutar comandos, los cuales no se ejecutarán hasta el momento cuando sea presionada la tecla Enter, además de poder ser visualizado mientras este mismo se es escrita. Siendo así los principales comandos los siguientes:

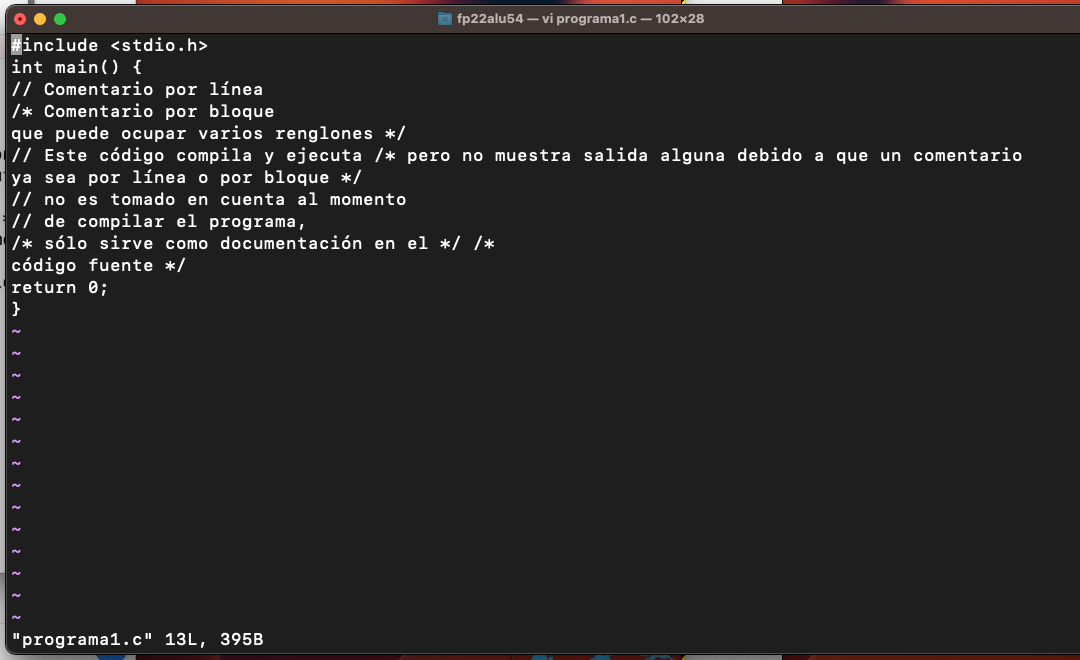
* :q para salir del vi sin haber editado el texto desde la última vez que se guardó
* :q! para salir del vi sin cuardar
* :w para guardar los cambios
* :wq para salir del vi y guardar los cambios

**Ejecución:**

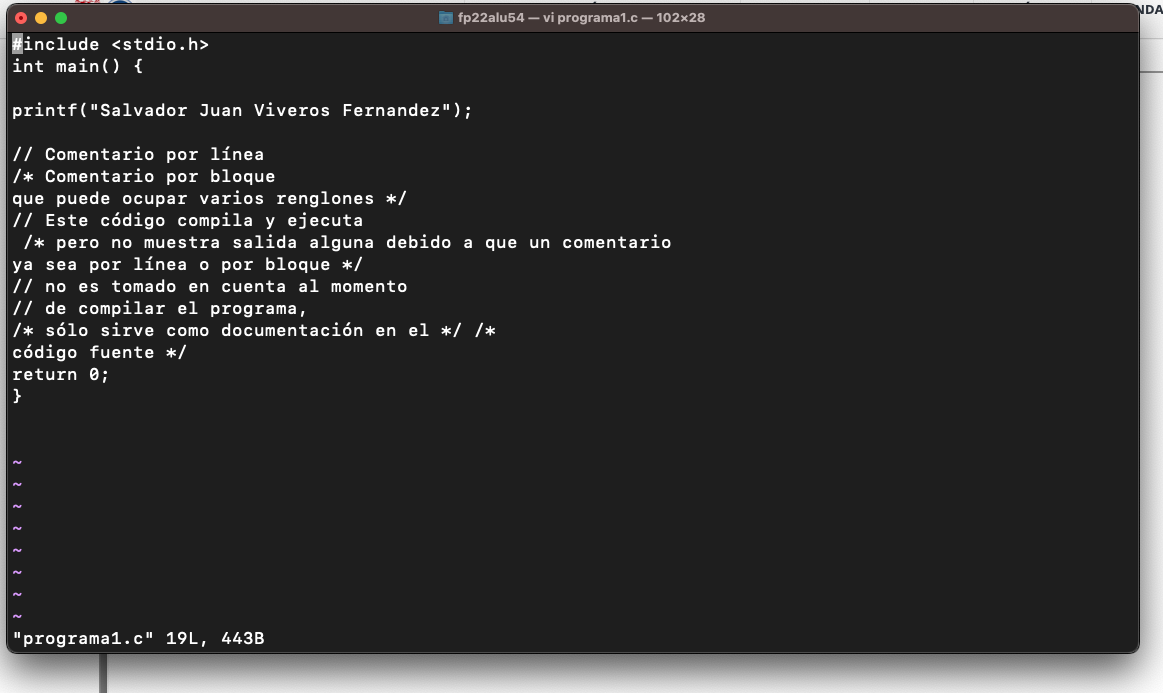
Una vez se haya compilado el programa, se puede ir al apartado de ejecución, donde se puede distribuir para equipos que ejecuten el mismo sistema operativo y tengan la misma plataforma de hardware, pudiendo ser ejecutado de diferentes formas dependiendo del sistema operativo que la máquina esté utilizando.

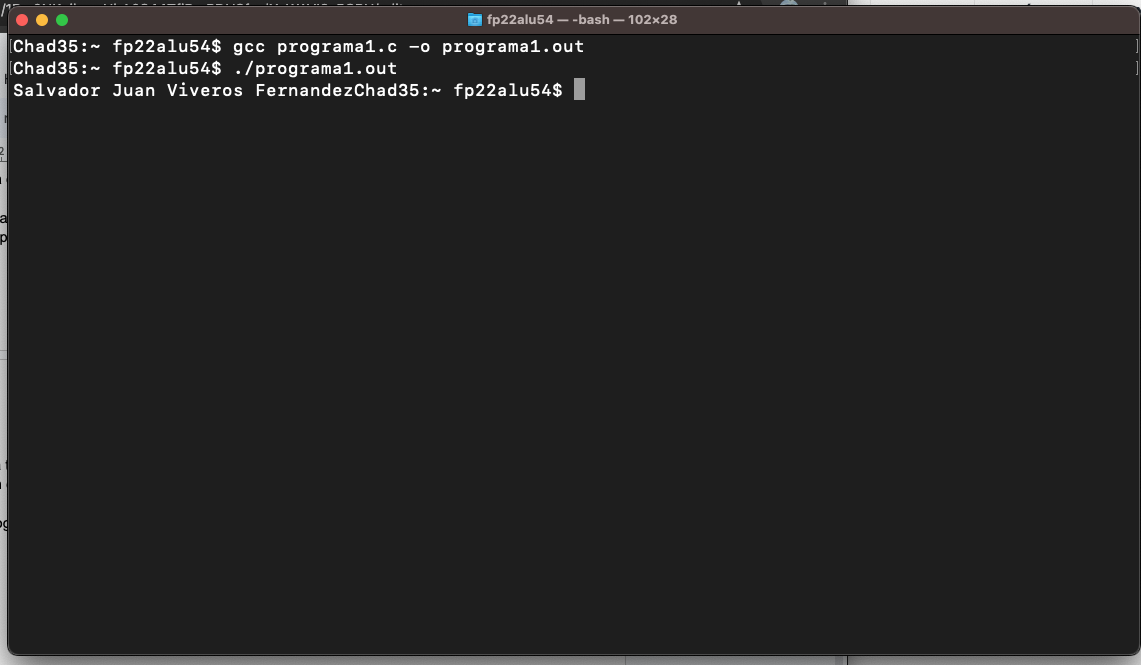
**Ejercicios:**

1. Se crea un programa, el cual al ser ejecutado no dará salida, puesto que los comandos sólo son una seguidilla de comentarios:

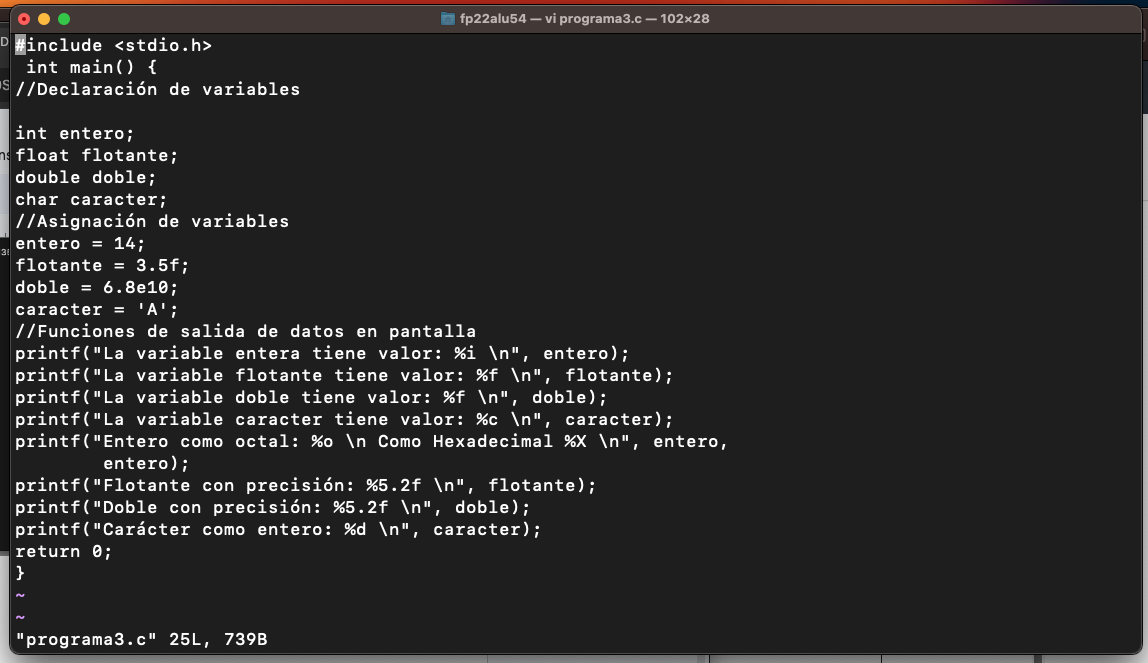


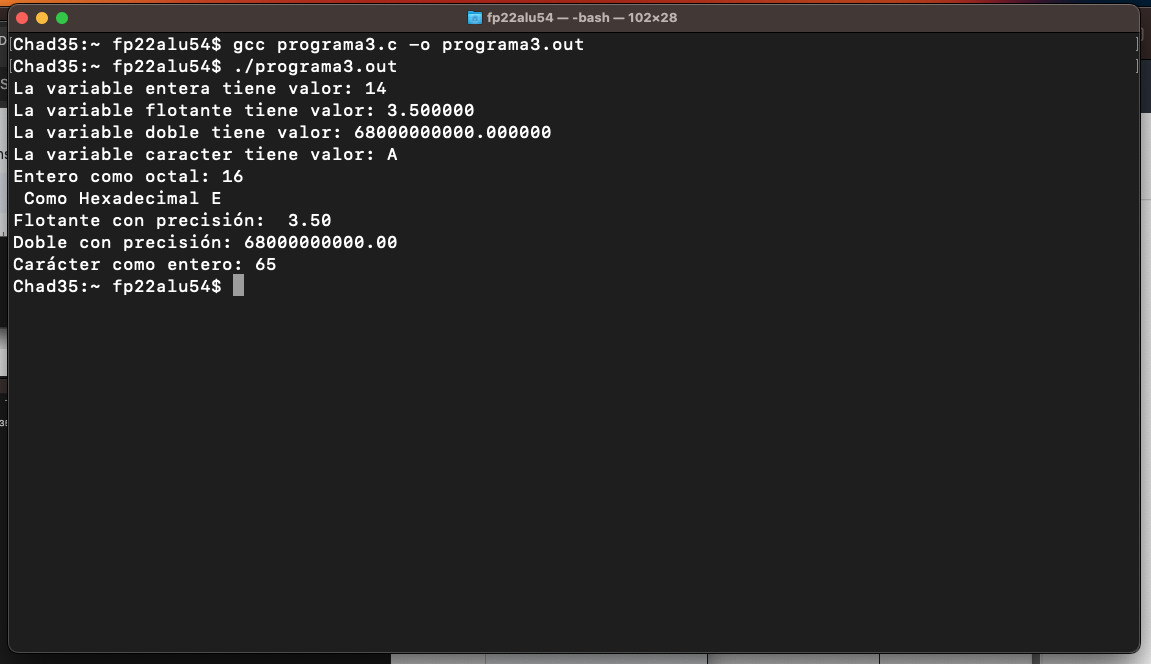
1. Se crea un programa que, después de ser compilado y ejecutado, puede imprimir mi nombre:



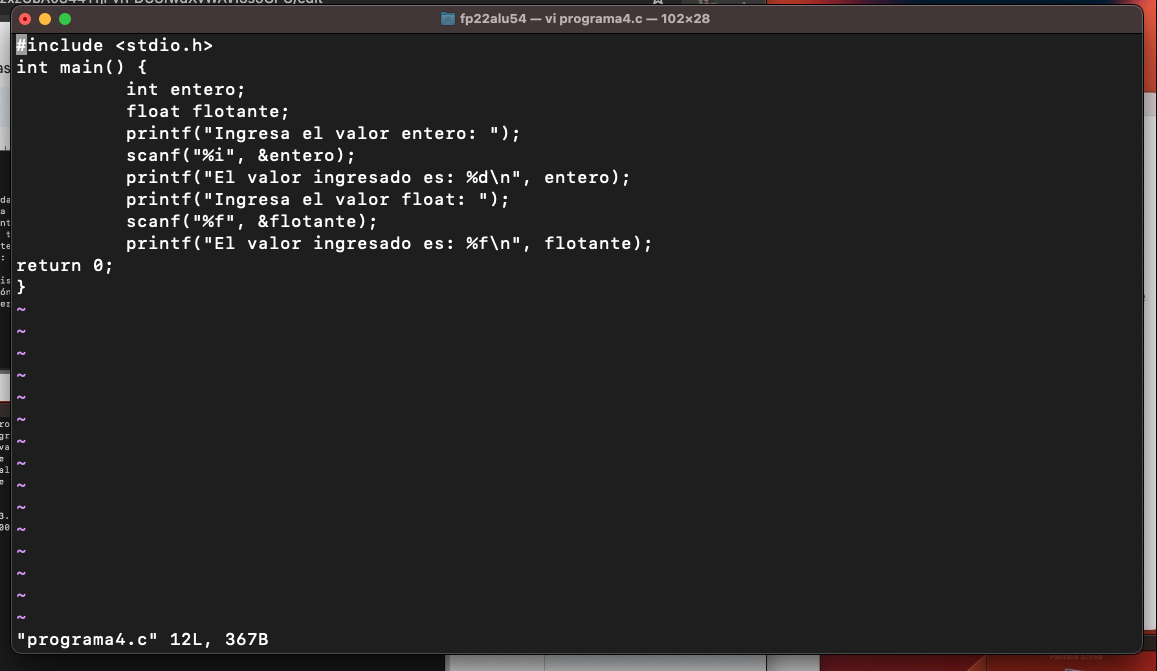


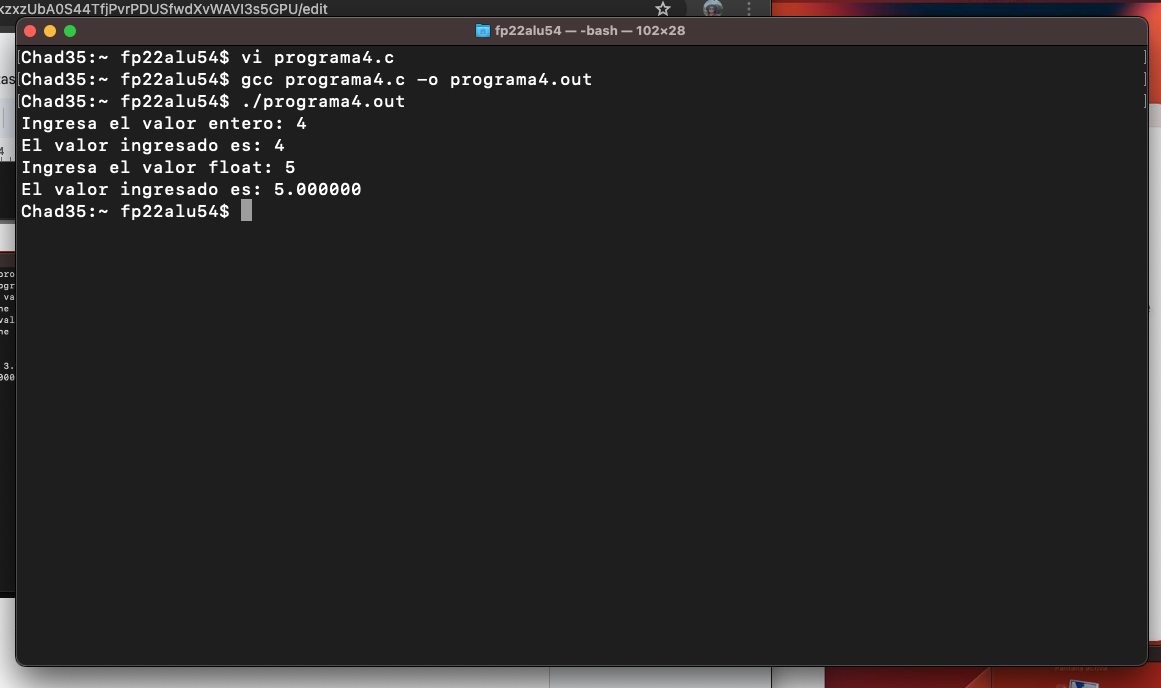
1. Se crea un programa, el cual, tras ser ejecutado, da información sobre valores que han sido previamente asignados en el propio código:





1. Se crea un programa que da el valor de diversos números con distintos formatos, siendo dichos números proporcionados por el usuario:





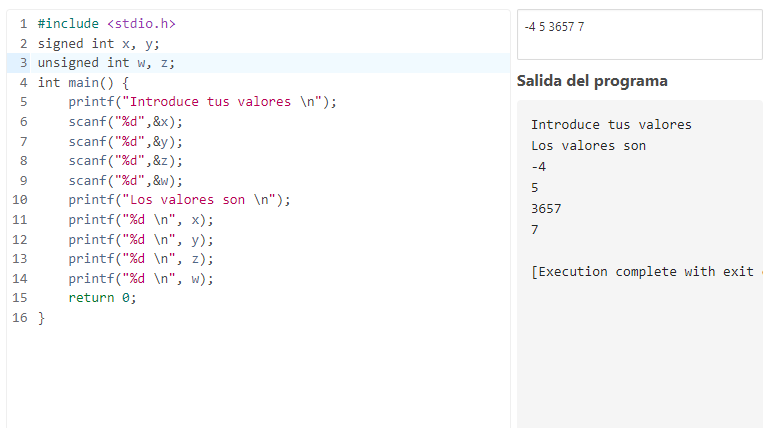
**Tarea:**

1. **Investigar cual es el dato que se encuentra por default con lenguaje C (signed o unsigned) y mencionar las características con las que debe crearse una variable.**

Si no se especifica si la variable creada es signed (con signo) o unsigned (sin signo), por defecto, la variable será almacenada como una variable signed, indicando que la variable puede llevar signo.

Las variables al ser declaradas con alguno de los calificadores, debe llevar el orden *Calificador Tipo de variable Nombre de la variable*.

1. **Crea un programa en el que declares 4 variables haciendo uso de las reglas signed/unsigned, las cuatro variables deben ser solicitadas al usuario (se esmplea scanf) y deben mostrarse en pantalla (emplear printf):**

****

1. **Comparación entre Editor de Texto y Procesador de Texto (Realizar una tabla comparativa):**

|  |  |
| --- | --- |
| **Procesador de texto** | **Editor de texto** |
| Se puede modificar la fuente además de otras características de las letras. | No admite ningún tipo de formato extra. |
| Permite utilizar imágenes y demás recursos. | No permite la introducción de imágenes. |
| Permite modificar la orientación del texto. | No permite modificar la orientación del texto. |
| Generalmente de almacenan en .docx | Generalmente se almacenan en .txt |
| El tamaño de los documentos depende de más factores. | El tamaño del texto depende de únicamente de la cantidad de caracteres. |

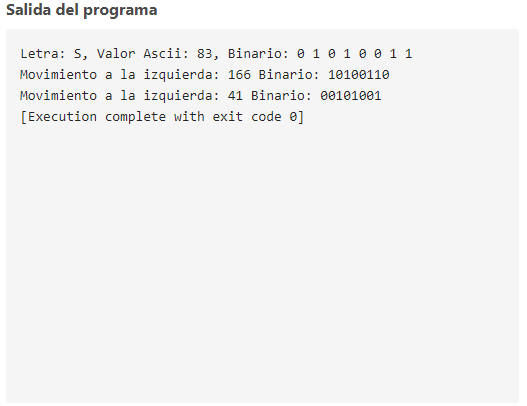
1. **Indica los comandos para compilar y ejecutar un programa en iOS o Linux:**

Para compilar un programa en Linux se debe utilizar el comando *gcc* ***programa****.c – o* ***programa****.out*.

Por su parte, para ejecutar un programa, se hace uso del comando .*/****programa.****out.*

1. **Genera un programa y ejecútalo en la interfaz que elijas con el número binario de tu letra inicial del nombre y realiza un corrimiento a la izquierda y uno a la derecha del bit más significativo:**

****

****

**Bibliografía:**

Adrián, D. R. H (2022, 22 agosto). Calificadores de Datos en C: signed, unsigned, short y long. Diseño Web akus.nethhtps://disenowebakus.net/calificadores-datos.php#google\_vignette

García E. et al. (2022) Manual de Prácticas del laboratorio de Fundamento de programación, recuperado de <http://lcp02.fi-b.unam.mx/>

Velasco, R (2023, 31 octubre). Qué diferencia entre un editor y un procesador de texto. SoftZone. https://www.softzone.es/2017/03/22/diferencia-editores-procesadores-texto /