



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Alejandro Pimentel Alarcón

*Asignatura:* Fundamentos de programación

*Grupo:* Bloque 135

*No de Práctica(s):* Práctica 6

*Integrante(s):* Partida Arias Emily Rachel

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:* 41

*Semestre:* 2020-1

*Fecha de entrega:* Lunes 30 de septiembre

*Observaciones:* Bastante bien

**CALIFICACIÓN:** 10

## Práctica 6. Entorno de C

### Objetivo:

Conocer y usar los ambientes y herramientas para el desarrollo y ejecución de programas en Lenguaje C, como editores y compiladores en diversos sistemas operativos.

### Introducción:

#### *¿Qué es el lenguaje C?*

Lenguaje de programación C. También conocido como “Lenguaje de programación de sistemas” desarrollado en el año 1972 por Dennis Ritchie para UNIX un sistema operativo multiplataforma. El lenguaje C es del tipo lenguaje estructurado como son Pascal, Fortran, Basic. Sus instrucciones son muy parecidas a otros lenguajes incluyendo sentencias como if, else, for, do y while. Aunque C es un lenguaje de alto nivel (puesto que es estructurado y posee sentencias y funciones que simplifican su funcionamiento) tenemos la posibilidad de programar a bajo nivel (como en el Assembler tocando los registros, memoria etc. ).



```
1 /* This line basically imports the "stdio" header file, part of
2  * the standard library. It provides input and output functionality
3  * to the program.
4  */
5 #include <stdio.h>
6
7 /*
8  * Function (method) definition. This outputs "Hello, world" to
9  * standard output.
10 */
11 void sayHello() {
12     // printf() in the specified text (with optional
13     // formatting options)
14     printf("Hello, world!\n");
15 }
16
17 /*
18  * This is a "main function". The compiled program will run the code
19  * defined here.
```

### Desarrollo:

**Actividad. Hacer una investigación muy somera acerca de los siguientes tipos de archivos:**

- txt
- markdown
- html
- LaTeX
- Csv

El término extensión del fichero (también, extensión del archivo) es una cadena de caracteres anexada al nombre de un archivo, habitualmente precedida por un punto. Su función principal es distinguir el contenido del archivo, de modo que el sistema operativo disponga del procedimiento necesario para ejecutarlo o interpretarlo. Sin embargo, la extensión es solamente parte del nombre del archivo y no representa ningún tipo de obligación respecto a su contenido.

## **Txt**

Documentos de texto plano guardados en formato TXT se pueden crear, abrir y editar utilizando una amplia variedad de programas de procesamiento de texto y de edición de textos desarrollados para sistemas Linux, ordenadores y plataformas Mac Microsoft basado en Windows.

## **Markdown**

Al utilizar la sintaxis de Markdown, serás capaz de producir archivos que pueden ser legibles como texto plano y que a la vez están listos para ser formados en otras plataformas. Muchos generadores de bitácoras y de sitios estáticos, así como sitios como GitHub, también aceptan Markdown y traducen estos archivos a HTML para su visualización en la web.

## **Html**

Esta extensión identifica archivos cuyo contenido está en lenguaje HTML. Este lenguaje es el utilizado para mostrar la información hipertextual en la Web.

## **LaTeX**

LaTeX es un sistema de preparación de documentos. Con él puedes preparar manuscritos, artículos de revista, cartas, tesis, presentaciones y cualquier tipo de documento que quisieras imprimir en papel o mostrar en pantalla.

## **Csv**

Los archivos CSV (del inglés comma-separated values) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas.

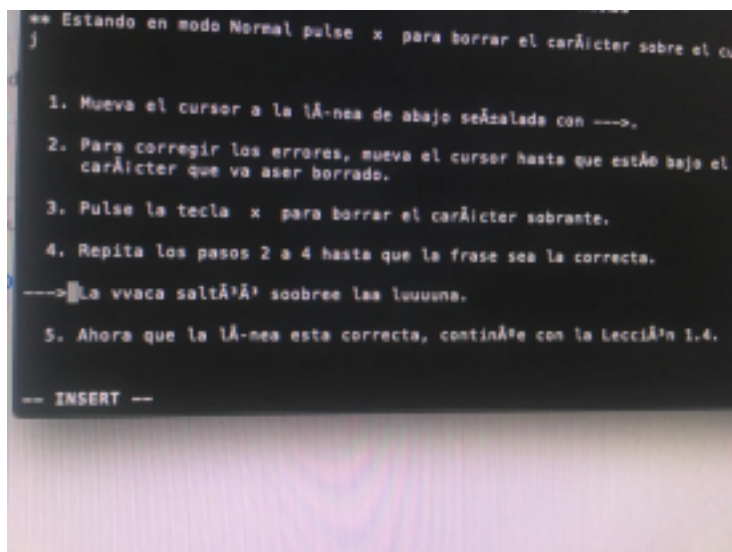
## **Actividad. Seguir el tutor de vim.**

Primero abrí el archivo que puso el profesor de vim y lo pegue en la terminal para empezar a seguir el tutor.

1.-Los primeros mandatos que te da el tutor es el de mover el cursor con las teclas **k** hacia arriba, **j** hacia abajo, **h** izquierda y **l** a la derecha.

2.-Para entrar y Salir de vim se escribe el mandato **:q!** enter sin guardar cambios, si quieres guardar cambios se escribe **:wq** enter, si quieres entrar al editor pones vim totor enter y vuelves a entrar.

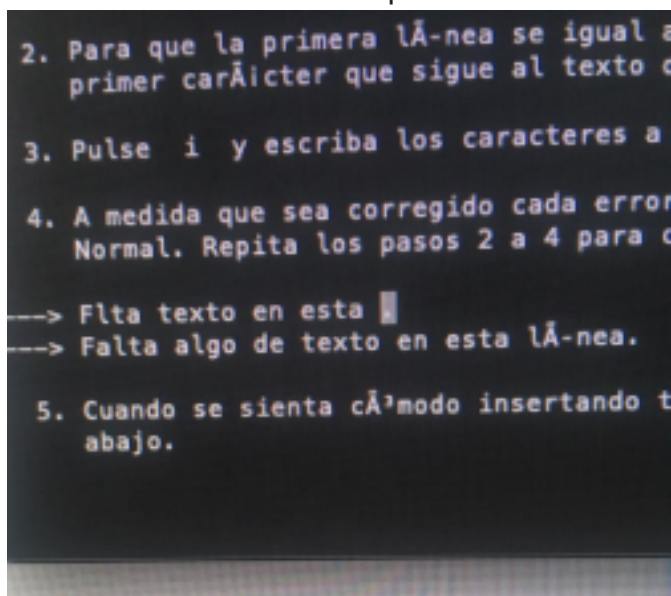
3.-Para borrar un carácter te pones en lo que deseas borrar con el cursos y oprimes la Tecla **x**:



---> La vaca salta sobre la luna.

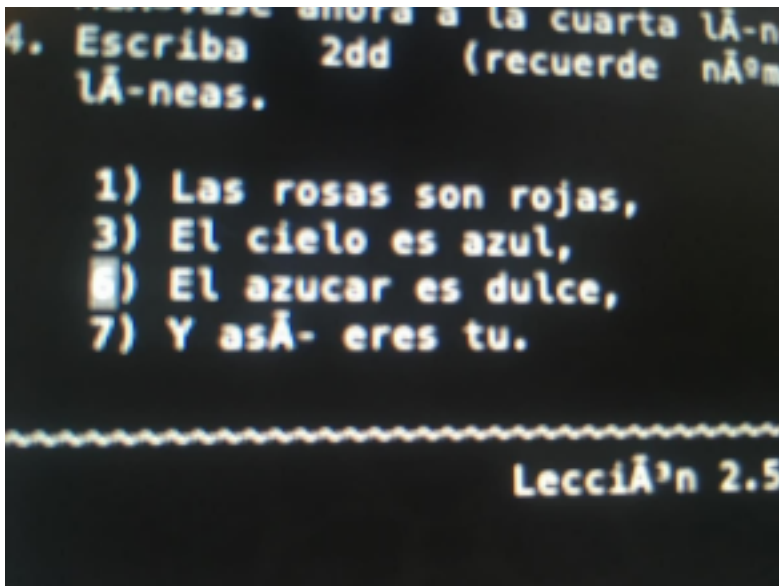
En este caso la oración estaba mal escrita pero con el cursor y x la puedes modificar.

4.-Para insertar texto en la posición del cursor se pulsa i y se escribe el texto



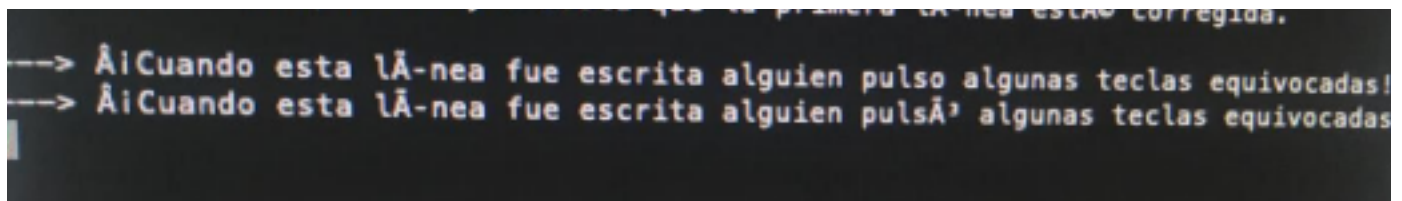
---> Falta algo de texto en esta línea.

5.-Para borrar una palabra, con el cursor te pones sobre la primer letra y oprimes **dw** para borrar, También puedes utilizar el mandato **d\$** para borrar una línea completa a partir del cursor, para borrar una línea entera se pulsa **dd**, para deshacer acciones previas se pulsa **u**.



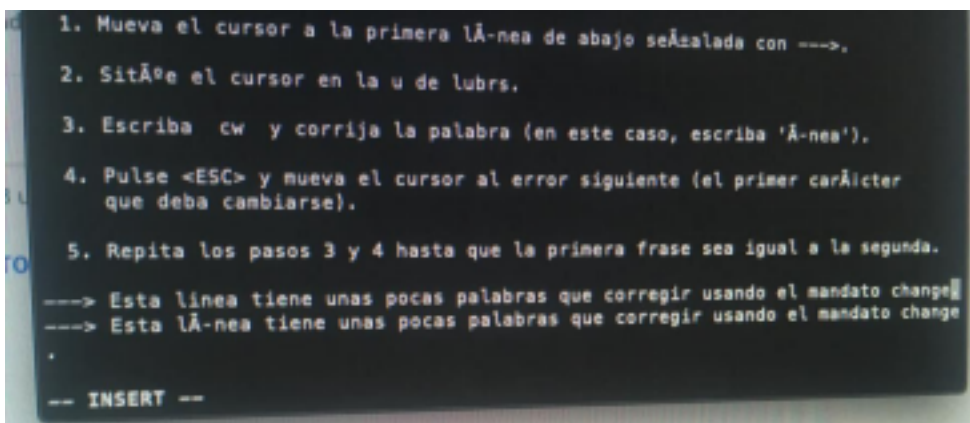
En este ejercicio borre la línea 2 con **dd** y con **2dd** borre las líneas 4 y 5

6.-Para sustituir algún texto que borramos pulsamos **p**, esto pone el texto borrado después del cursor, para sustituir un carácter pulsamos **r** y luego el nuevo carácter que queremos,

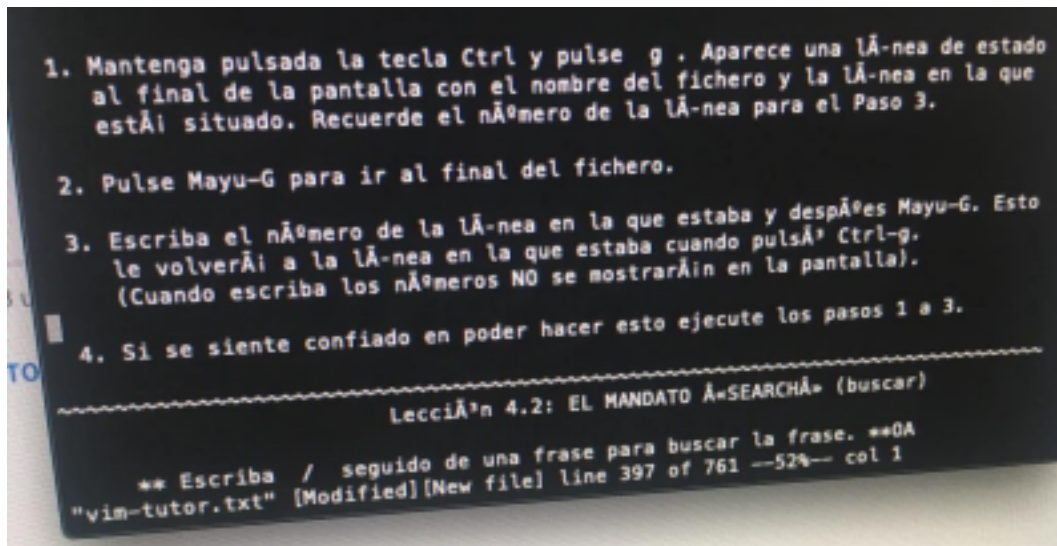


En este caso se puso el cursor en el carácter que se deseaba y para sustituir los caracteres pulsamos **r** y escribimos en nuevo carácter.

7.-Para corregir una palabra, nos ponemos en el cursor en la segunda letra de la palabra errónea, pulsamos **cw** y corregimos la palabra, siempre pulsando **esc** para saber que estamos ahí

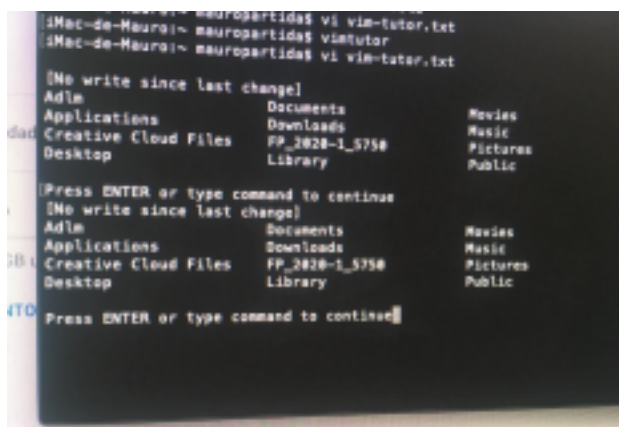
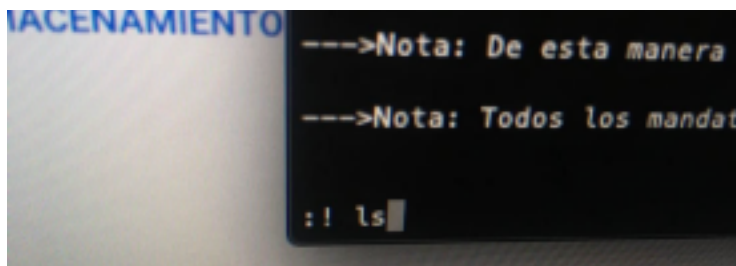


8.-Con el mandato **ctrl g** nos muestra la posición del cursor en el fichero y su estado, al oprimir en mayúscula la letra **G** el cursor se mueve al final del fichero si pones el numero de la línea en la que estabas y de nuevo **G** te manda al fichero y en la posición en la que estabas.



En este caso estamos situados en la línea 397 y al presionar **G** nos manda al final del cursor, si volvemos a poner 397 y **G** nos manda a la línea en la que estábamos.

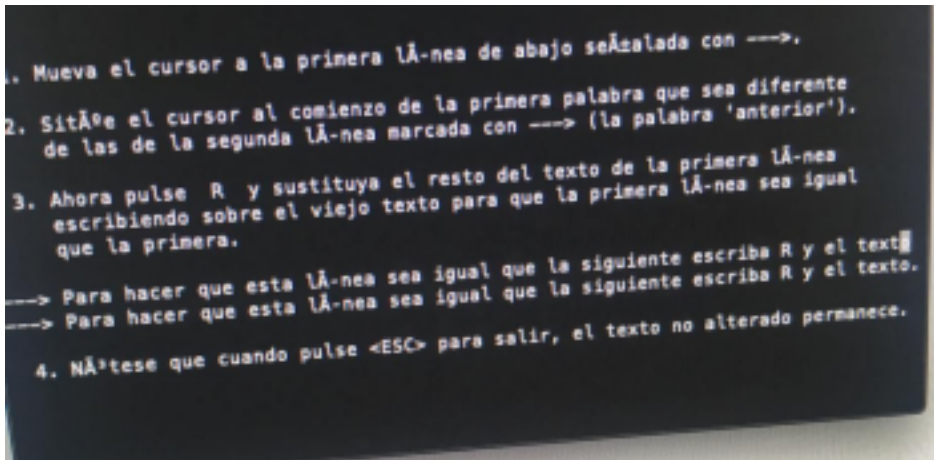
9.-El mandato **:! Ejecuta un mandato externo**, si pulsamos **:! ls** nos manda al contenido del directorio, con **:#, #w Nombre\_de\_fichero** guarda desde las líneas **#** hasta la **#** en el fichero.



En este caso oprimi **:! ls** y me mando al directorio de la computadora, con enter regresas a donde estabas.

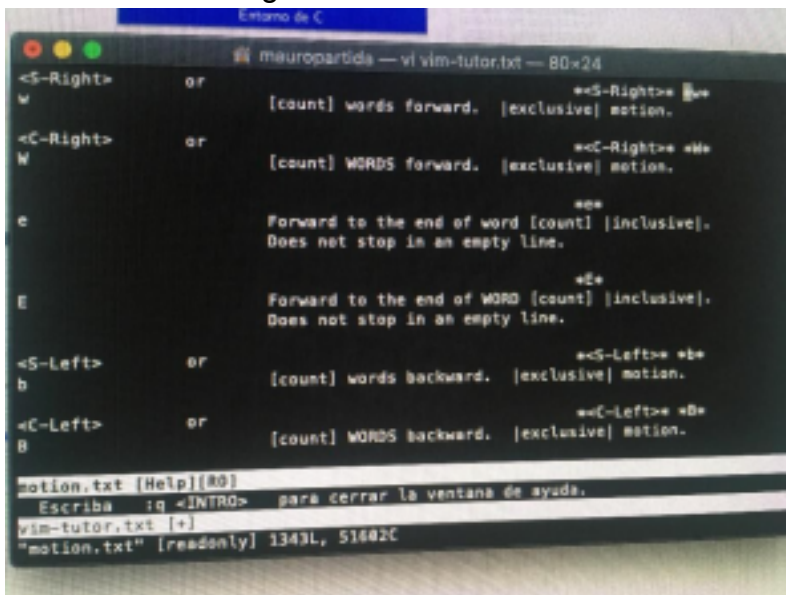


10.-Si pulsamos o abre una línea abajo del cursor y pulsando **O** se abre una línea sobre la que está cursor, pulsando **a** insertamos texto después del carácter somber el cursor, pulsando **R** se entra en modo reemplazar y para salir **esc**.



En este ejercicio al pulsar **R** sobre la palabra que era diferente se escribió la de la línea de abajo y se reemplazó por la que estaba.

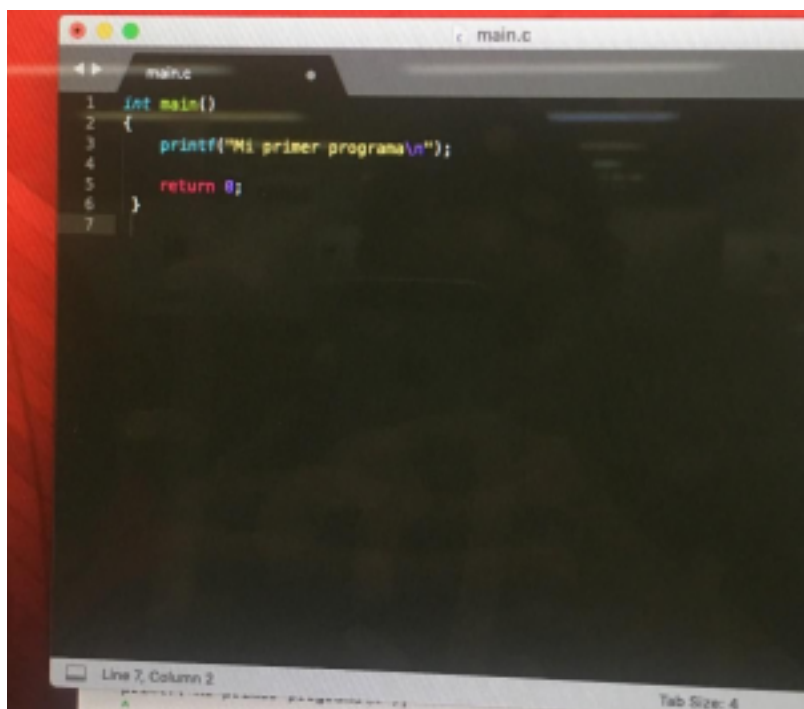
11.-Si necesitamos ayuda solo escribimos el mandato :help seguido de un argumento, si escribimos el mandato :help w , :help c\_<T y enter en los dos casos nos manda a la ayuda que buscamos en el argumento.



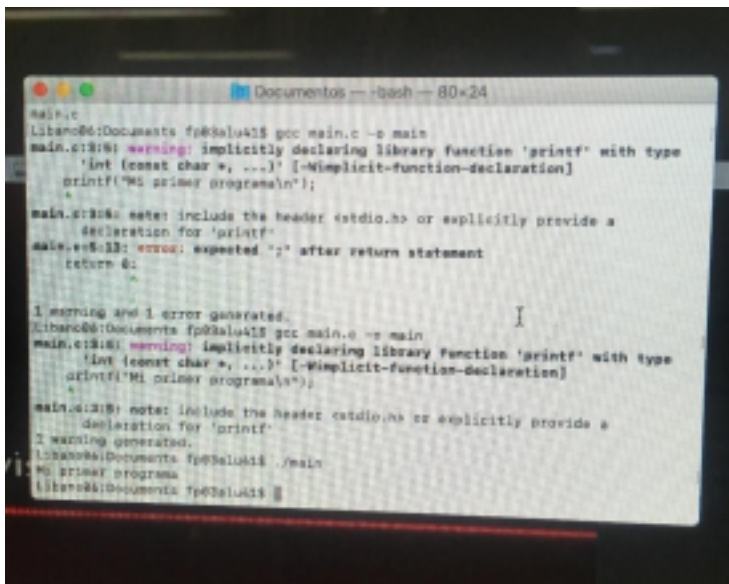


## Sublime text

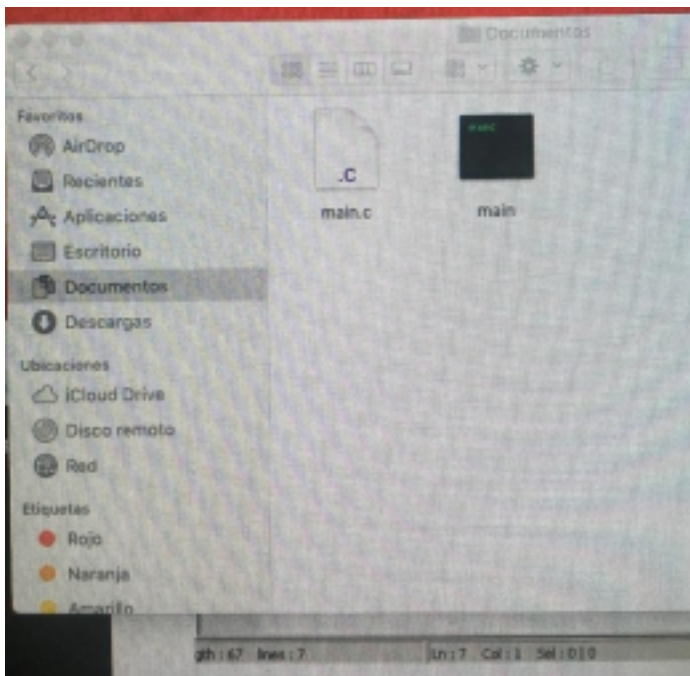
En las macs de la escuela abrimos el editor de texto Sublime y escribimos lo que se indicaba, pero antes guardando el archivo con el nombre **main.c** y con el comando **gcc main.c -o main** se compilaba el programa a un código ejecutable por la máquina o sea C.







```
Libero@66:Documentos fp@3alu415 $ gcc main.c -o main
main.c:1:16: warning: implicitly declaring library function 'printf' with type
'int (const char *, ...)' [-Wimplicit-function-declaration]
printf("Mi primer programa\n");
^
main.c:13:6: error: expected ';' after return statement
return 0;
^
1 warning and 1 error generated.
Libero@66:Documentos fp@3alu415 $ gcc main.c -o main
main.c:1:16: warning: implicitly declaring library function 'printf' with type
'int (const char *, ...)' [-Wimplicit-function-declaration]
printf("Mi primer programa\n");
^
main.c:13:6: note: include the header <stdio.h> or explicitly provide a
declaration for 'printf'
1 warning generated.
Libero@66:Documentos fp@3alu415 $ ./main
Mi primer programa
Libero@66:Documentos fp@3alu415 $
```



## Conclusion:

Para concluir con esta práctica, al estar siguiendo el tutor de vim es importante saber que siempre debemos pulsar esc para situarnos en modo normal y poder hacer los mandatos correctamente, también es crucial no aprendernos los mandatos de memoria sino ir practicando, el mismo tutorial al final dice que estos son algunos de los muchos mandatos que utiliza Vim así que solo es un tutor muy breve.

## Referencias:

- [https://www.ecured.cu/C\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://www.ecured.cu/C_(lenguaje_de_programaci%C3%B3n))
- <https://es.wikipedia.org/wiki/Markdown>
- <https://ondiz.github.io/cursoLatex/Contenido/03.DocumentoBasico.html>
- [https://es.wikipedia.org/wiki/Valores\\_separados\\_por\\_comas](https://es.wikipedia.org/wiki/Valores_separados_por_comas)