|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **Apuntadores** | | | **No.** | **15** |
| **Asignatura:** | **Metodos Numéricos** | **Carrera:** | **ISIC** | **Duración de la práctica (Hrs)** | **5** |

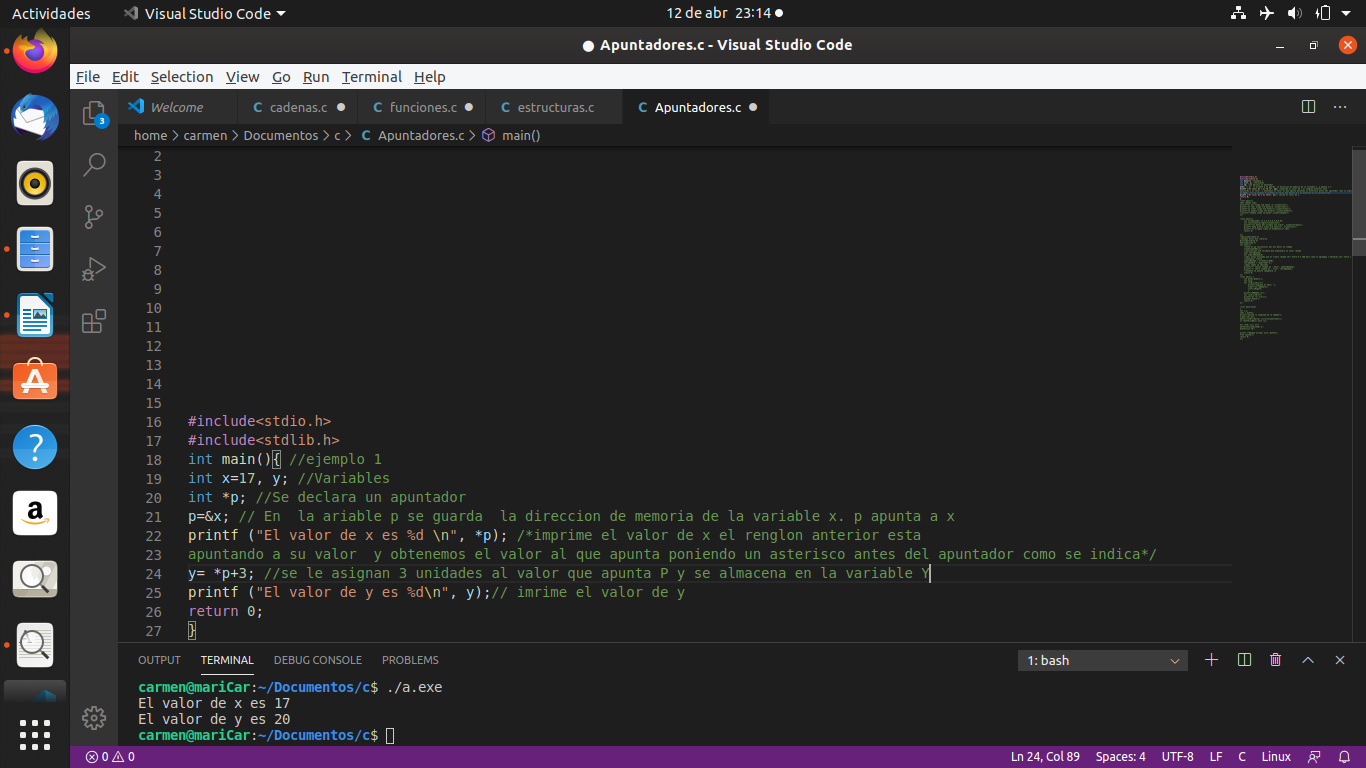
**I. Competencia(s) específica(s):\*\*\*\***

**II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Casa**

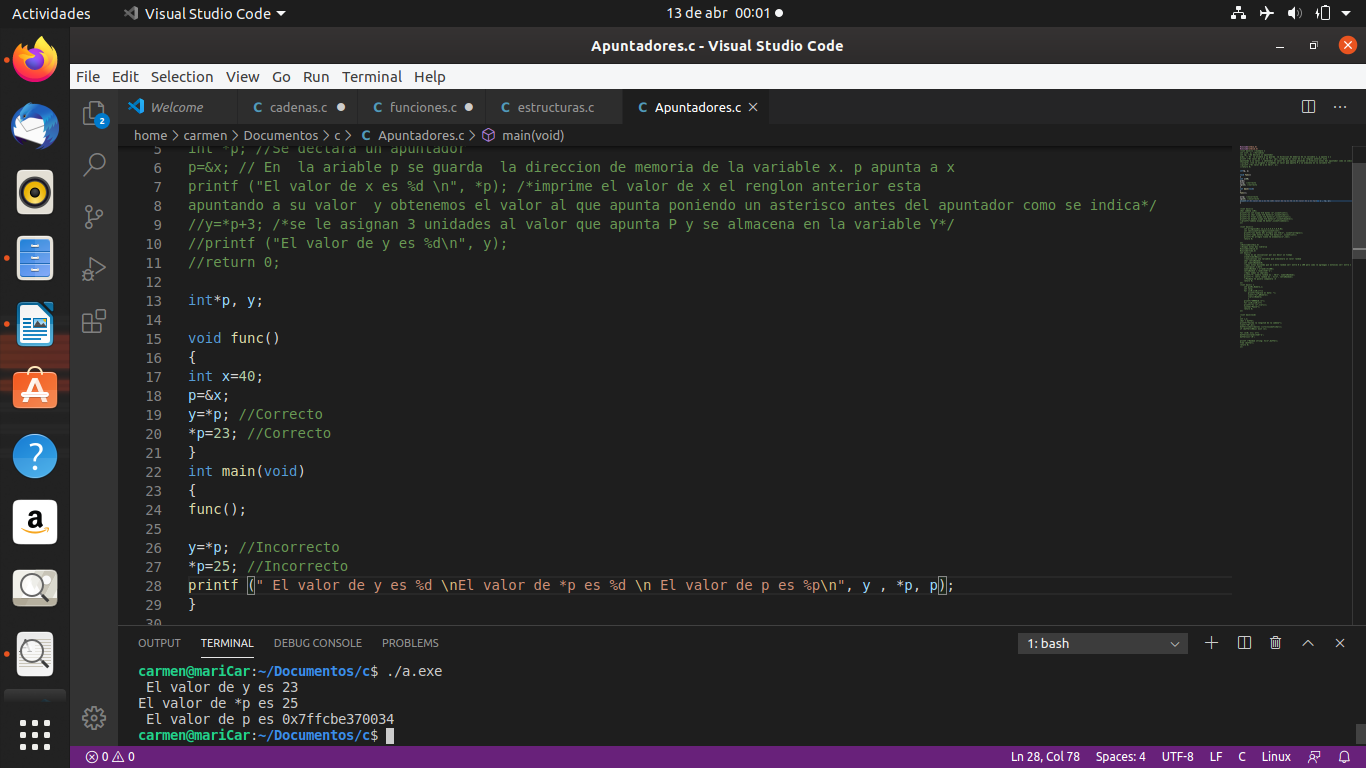
**III. Material empleado: Laptop con SO Linux, software Visual Studio Code.**

**IV. Desarrollo de la práctica:**

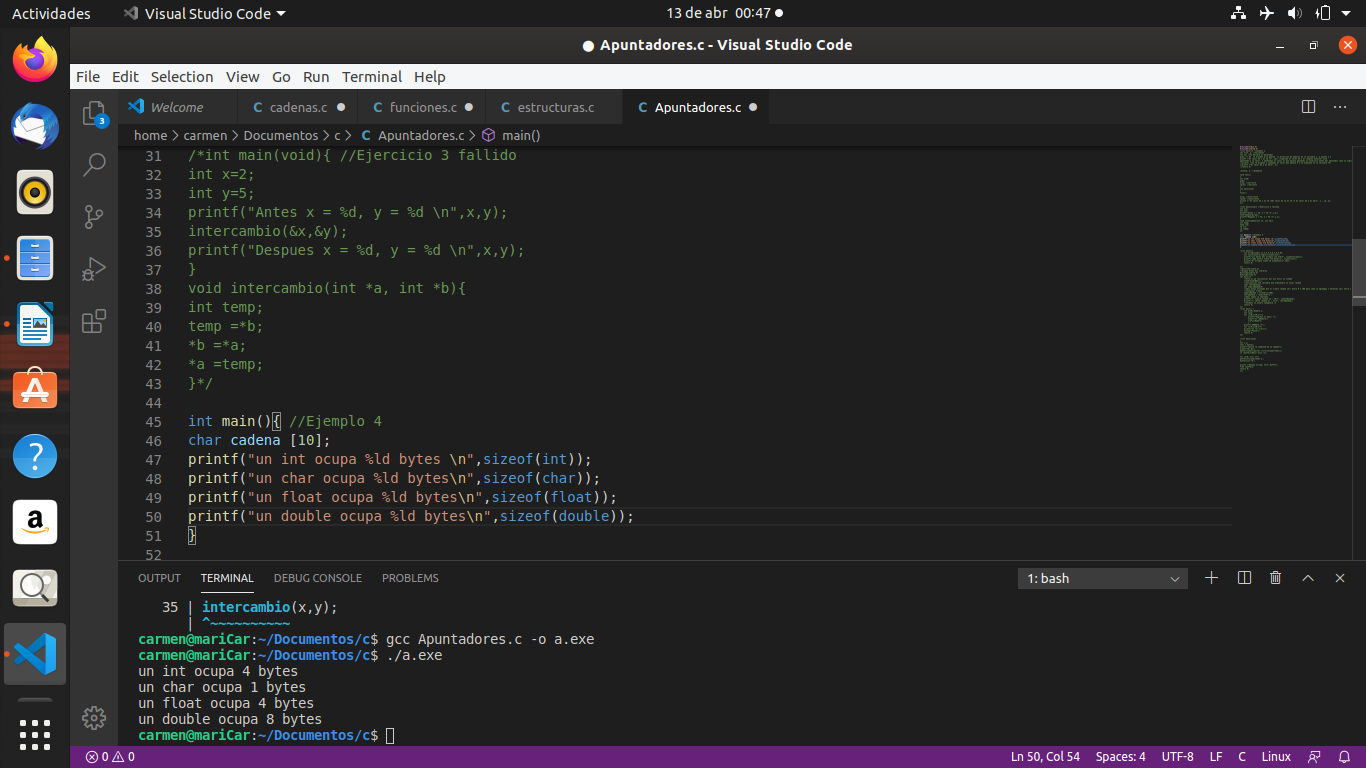
1. Primer ejemplo de apuntadores. Nos muestra la funcionalidad de los valores que obtiene una variable al utilizar un apuntador y forma de modificar el valor.



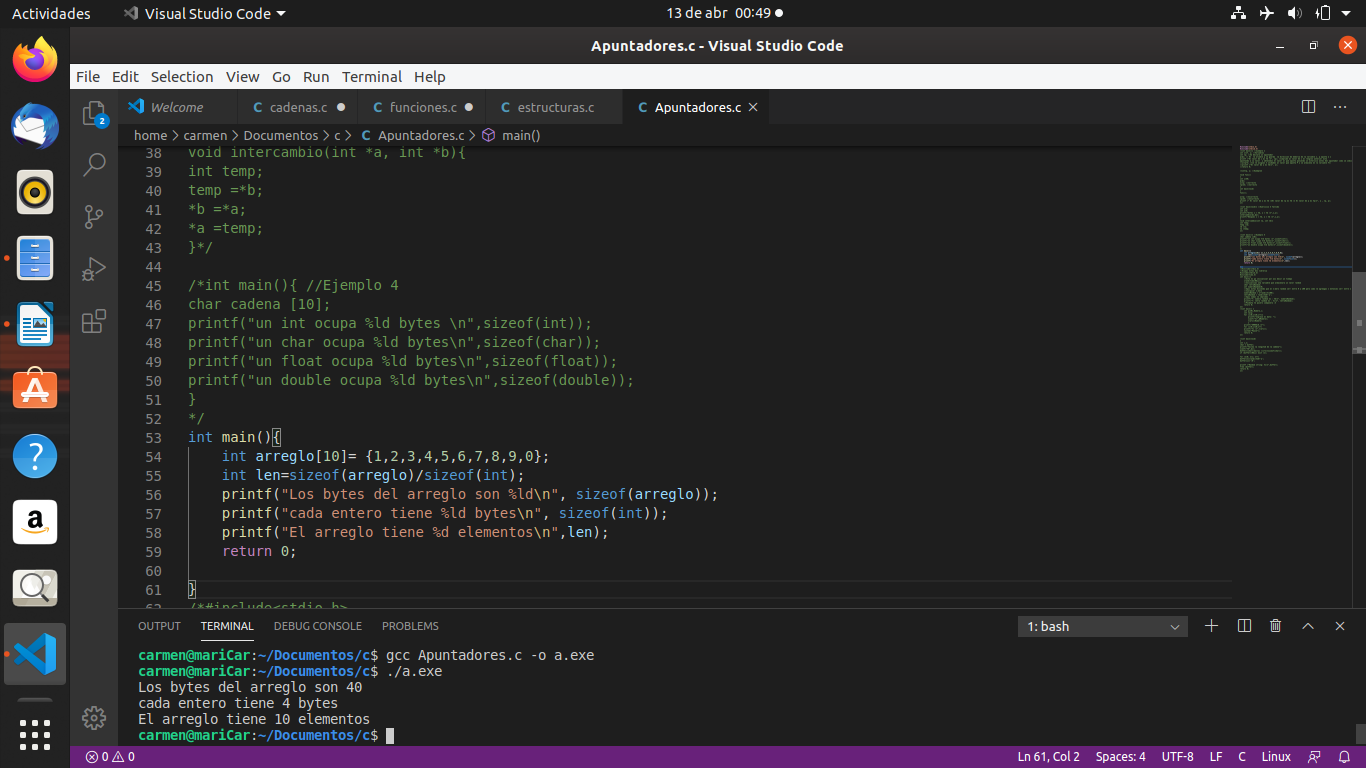
1. Utilizamos una función para enviar parametros como apuntadores y en cierta funcion puedan ser modificados sus valores de las variables a las que apuntan

****

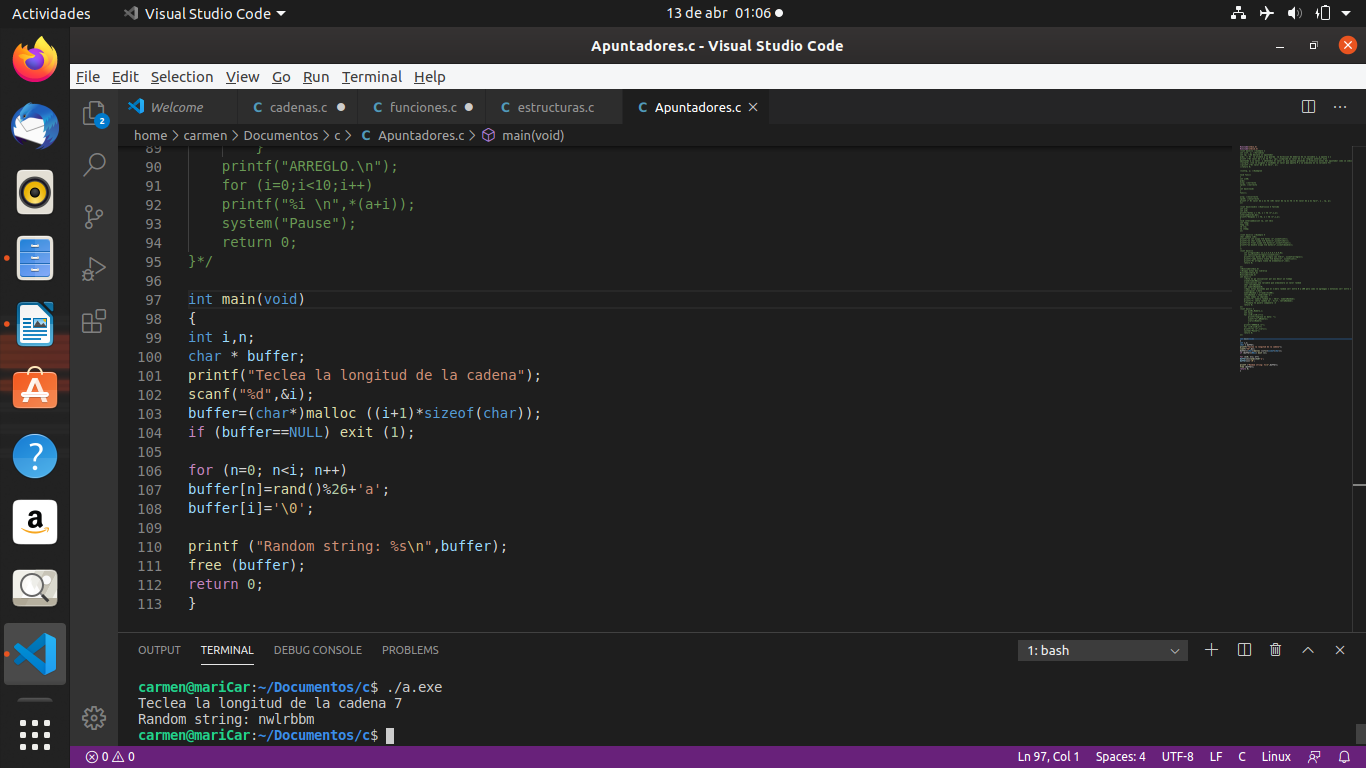
1. **Sizeof** nos ayuda a obtener la cantidad de bytes que ocupa cada tipo de dato

****

1. Obtenemos la cantidad de bytes que se ocupan en un arreglo de tipo entero con ayuda de **sizeof()**

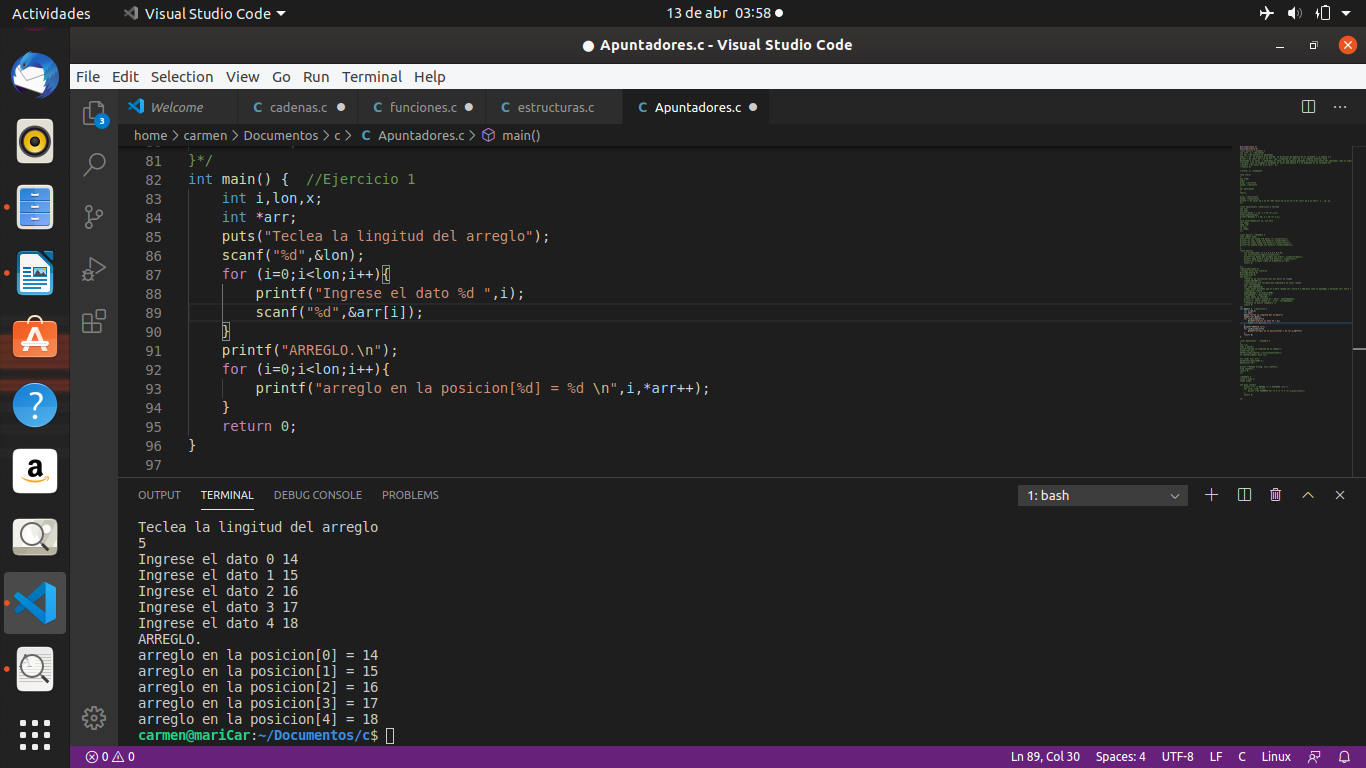
****

1. **Ocupamos un método llamado free(), molloc() y rand().**

****

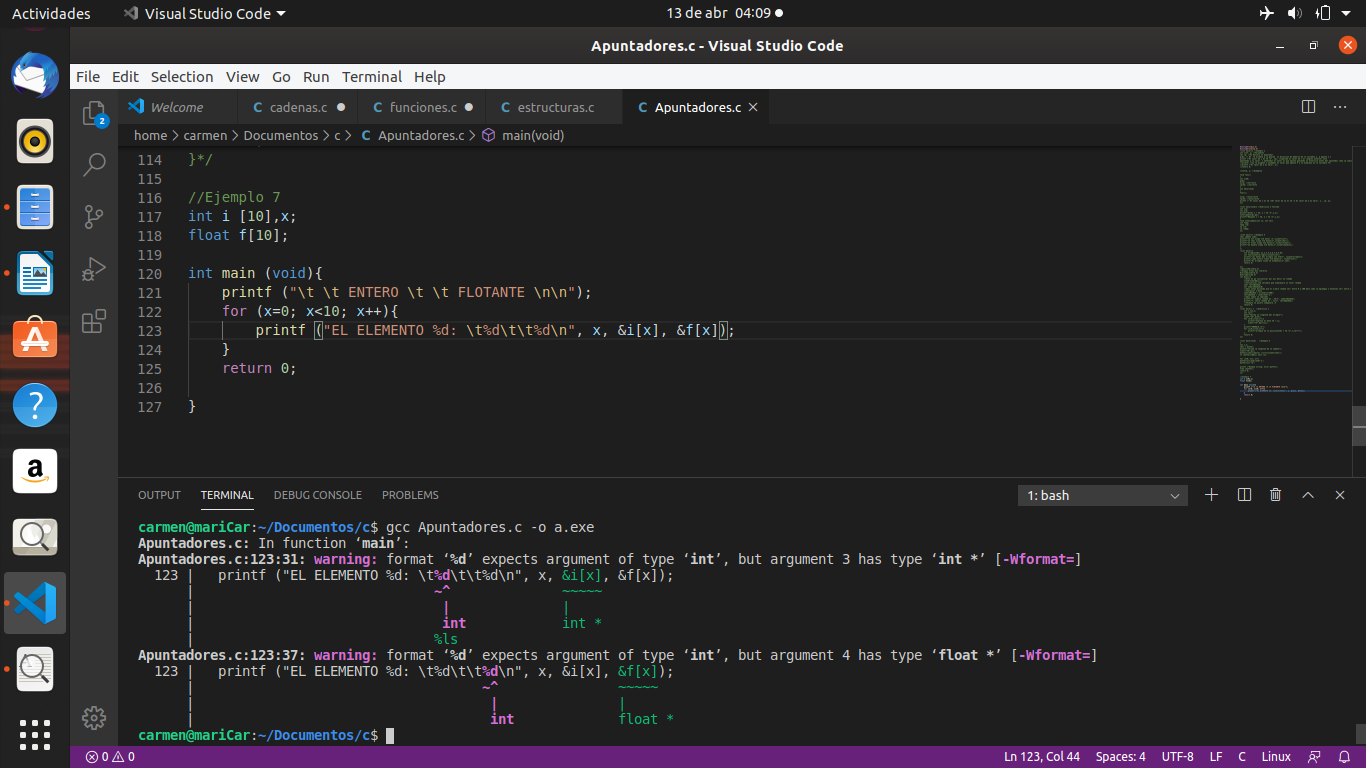
* **free():** Permite liberar la memoria reservada atraves de un apuntadores
* **molloc():** Reserva memoria sin importar el tipo de dato que almacena en ella.
* **Sizeof:** Obtiene la cantidad de bytes que ocupa un tipo o una variable en memoria.

1. Crea un arreglo entero de tamaño x, en donde x es ingresado por teclado.Llena todos los elementos del arreglo con datos ingresados por el usuario. Muestra los valores

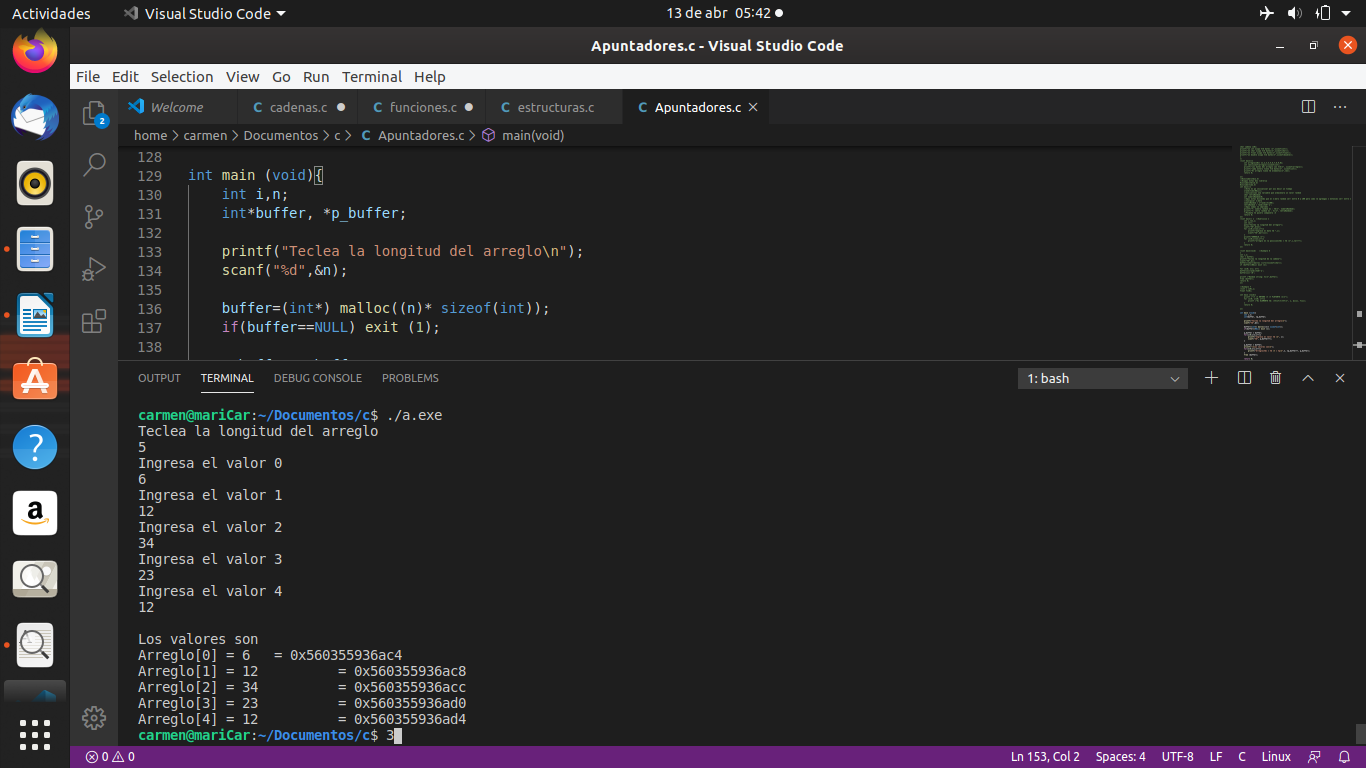
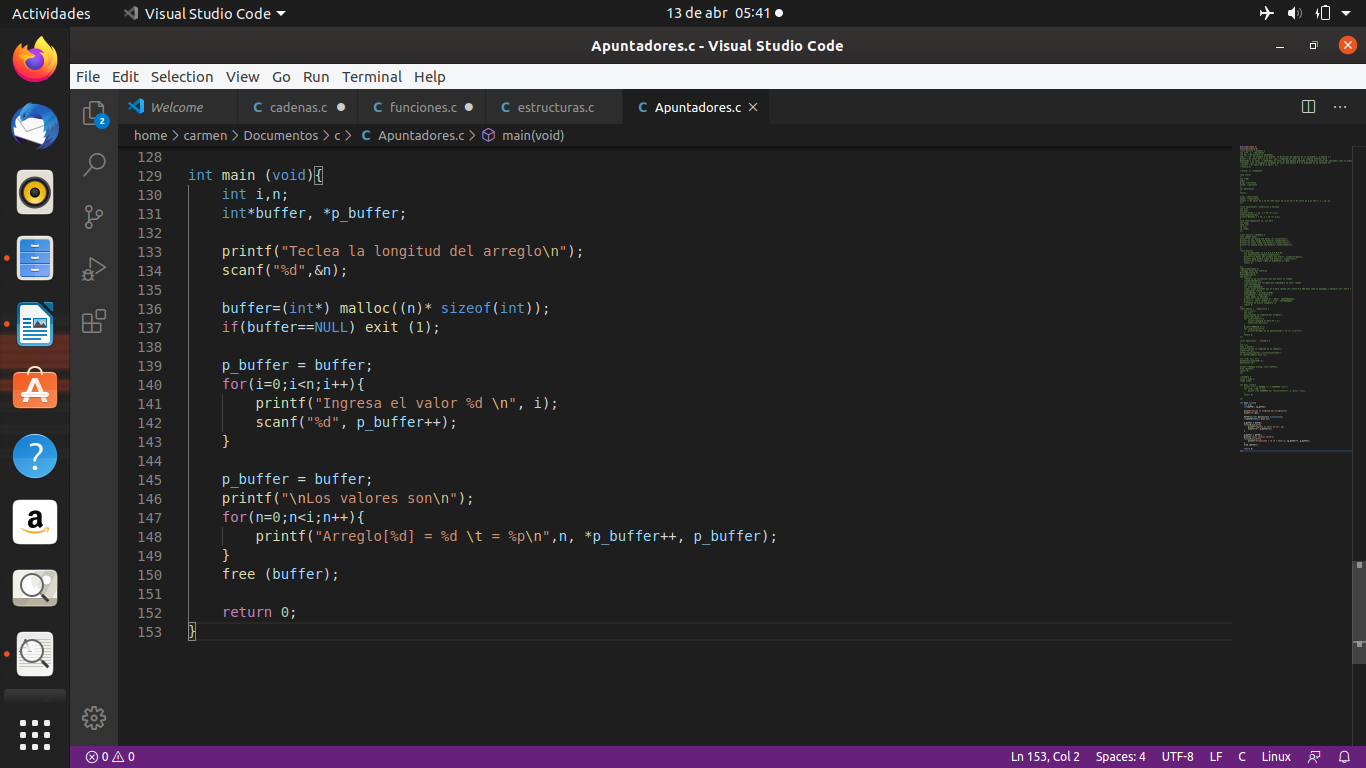
****

* declaramos un apuntador que sera utilizado como arreglo y 2 variables de tipo entero
* Utilizamos un for para para hacer las iteraciones de la cantidad de datos que se ingresaran en el arreglo. Dicha cantidad sera ingresada por el usuario también
* A la hora de pedir el Dato sera almacenado directamente en el apuntador utilizando la variable i para que el apuntador guarde el dato en la posición que indica i.
* finalmente imprimimos el arreglo haciendo una iteración con un ciclo for. El apuntador tendrá un incremento de una unidad por iteración que apuntara a cada uno de los valores almacenados y los imprimirá.

1. Los valores enteros y flotantes de las posiciones del arreglo



1. Crea un arreglo entero de tamaño x, en donde x es ingresado por teclado. Llena todos los elementos del arreglo con datos ingresados por el usuario

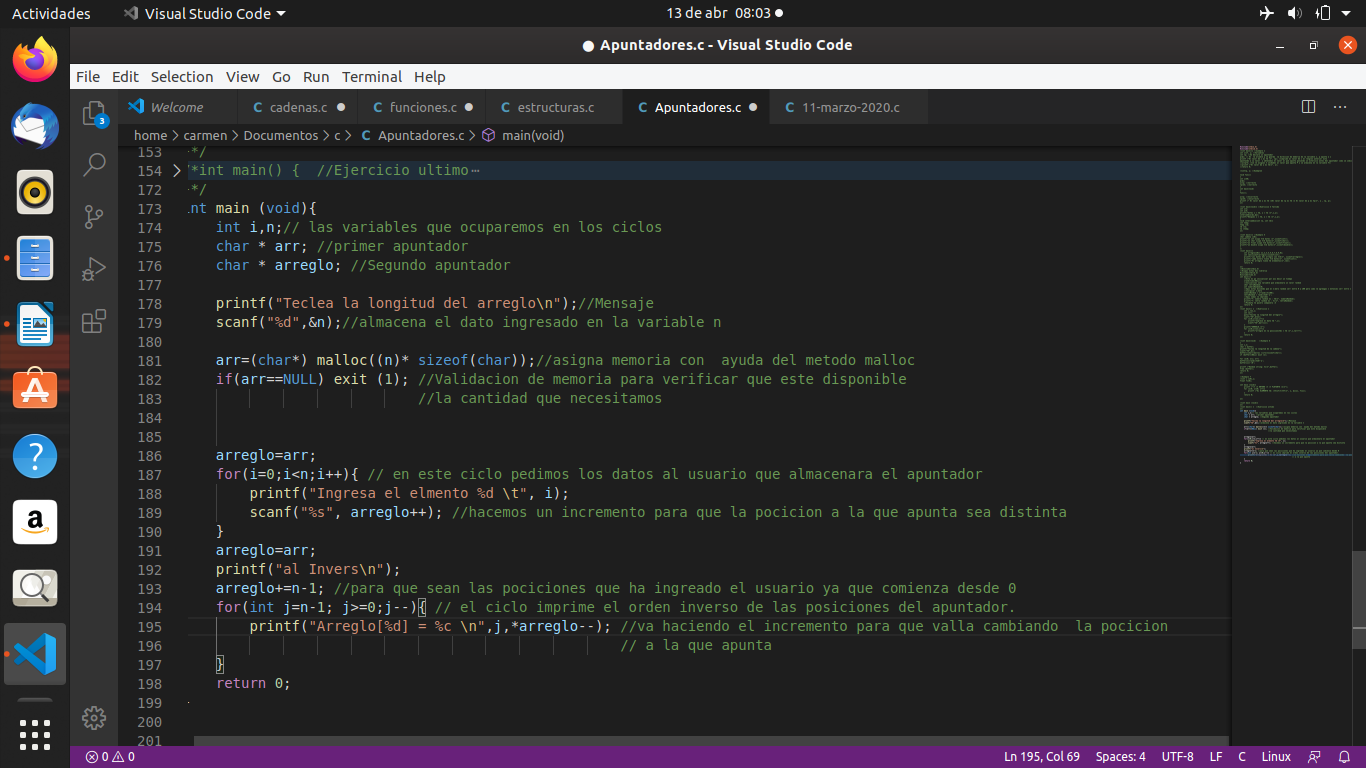


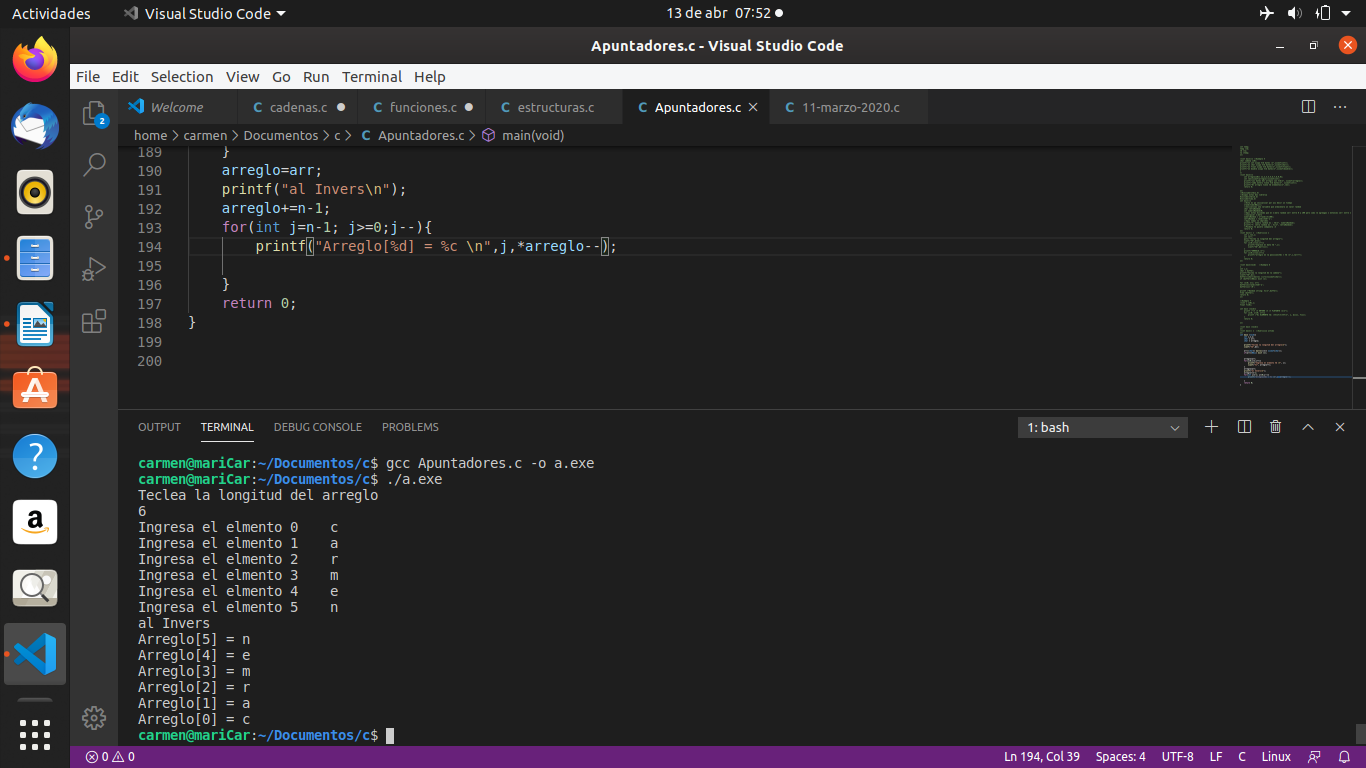
1. Crea un arreglo de tipo char de tamaño x, en donde x es ingresado por teclado.

Llena elemento por elemento del arreglo con letras ingresados por el usuario.

Muestra el arreglo impreso en forma inversa.

Todo debe ser manejado con apuntadores.





**V. Conclusiones:**

El uso de apuntadores es una herramienta que nos facilita ya que podemos utilizarlo como arreglos sin necesidad de declarar uno ya que la función de apuntar a diferentes valores es muy útil. Otros métodos que conocimos fueron free(), molloc() y Sizeof().

Este tipo de practicas nos ayudan bastante ya que conocemos distintas formas de implementación de las herramientas que ya hemos visto antes. Ademas de dominar un poco mas estos temas nuevos y así encontremos pronta solución a los problemas que mas adelante encontremos y la codificación cada vez sea mas corto.