|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **Apuntadores** | | | **No.** | **15** |
| **Asignatura:** | **Metodos Numericos** | **Carrera:** | **Ingenieria en sistemas computacionales** | **Duración de la práctica (Hrs)** |  |

**I. Competencia(s) específica(s):**

**Encuadre con CACEI: Registra el (los) atributo(s) de egreso y los criterios de desempeño que se evaluarán en esta práctica.**

|  |  |  |  |
| --- | --- | --- | --- |
| **No. atributo** | **Atributos de egreso del PE que impactan en la asignatura** | **Criterios de desempeño** | |
|  |  | **1** |  |
| **2** |  |
| **3** |  |
|  |  | **1** |  |
| **2** |  |
| **3** |  |

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

* **casa**

**III. Material empleado:**

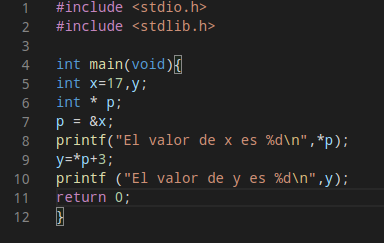
* **Laptop**
* **Visual studio code**

**IV. Desarrollo de la práctica:**

**Pdf 15 Apuntadores:**

**Programa 1: programa de referenciacion.**

**Progrma:**



**Linea 5: se declaran varias variables de tipo entero.**

**Linea 6: se declara un puntero der tipo entero y su setencia es el tipo de dato del puntero luego un asterisco y el nombre del puntero**

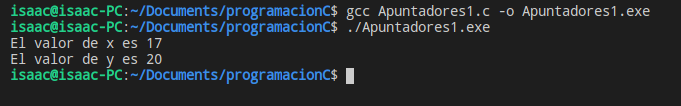
**Linea 7: al puntero se le asigna el valor en que esta almacenado x en la memoria ram.**

**Linea 8: imprime el valor que almacenaba x.**

**Linea 9: se la suma lo que guardaba el puntero p que es 17 y le suma 3 valores y lo almacena en y.**

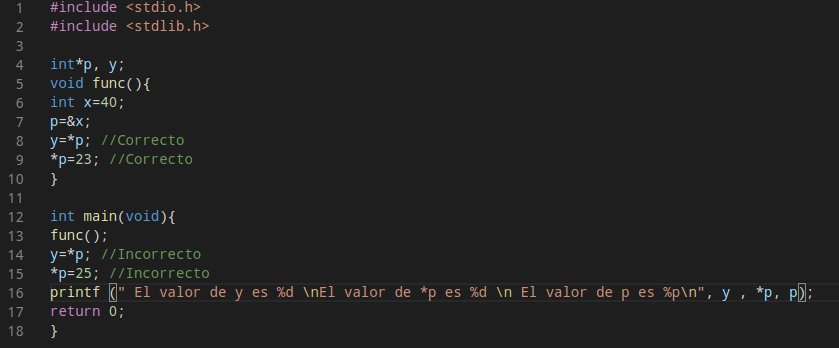
**Linea 10: imprime el valor que guarda y.**

**Ejecucion:**



Programa 2: referenciacion de punteros.

Programa:



Linea 4: se deeclaran un puntero y una variable.

Linea 13: se va a la funcion func.

Linea 6: se crae una variable en la funcion que almacena 40.

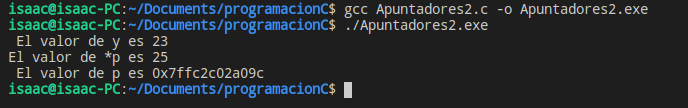
Linea 7: A p se le asigna la direcion de memoria de x.

Linea 8: en y se guarda el valor que tenia almacenado x.

Linea 9: y a x se le asigna el valor de 23

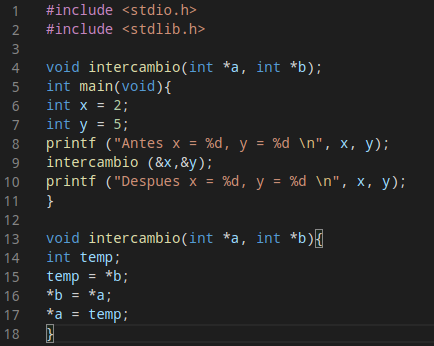
Linea 16: se imprime los valores.

Ejecucion:



Programa 3:paso de parametros por referenciacion.

Codigo:

n

Linea 9: se crea una funcion. Llamada intercambio con parametro dos punteros

Linea 6: se crea una variable que almacena a 2.

Linea 7: se crae otra variable que almacena a 5.

Linea 8: se imprimen los valores de cada variable.

Linea 9: se pasan los valores a la funcion intercambio.

Linea 14: se crea una variable en la funcion llamada temp.

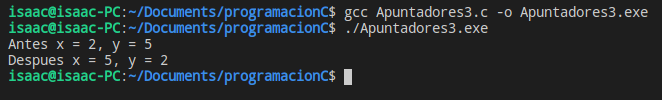
Linea 15: se guarda el valor de el puntero b en temp.

Linea 16: se guarda el valor del puntero a en el puntero b.

Linea 17: al puntero a se le asignan los valores que guardan temp.

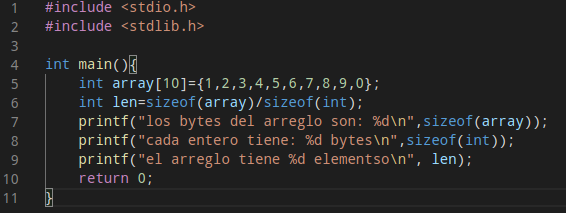
Linea 19: se imprime los valores cambiados de las variables.

EJECUCION:



Programa 4: la funcion sizeof.

Codigo:

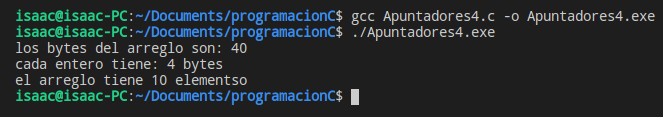


Linea 5: se crea un arreglo de 10 posiciones que ya almacena elementos.

Linea 6: se crea una variable llamada len con la funcion sizeof aplicada al array y al entero.

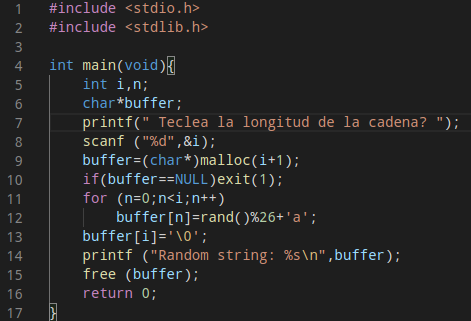
Linea 7-9: se imprimen los valores de cada variable.

Ejecucion:



Programa 5: programa de reservacion de memoria dinamica.

Codigo:



Linea 5: crean dos variables de tipo entero.

Linea 6: se crea un apuntador de tipo char.

Linea 7: se mannda un mensaje que ingrese la lomgitud de la cadena.

Linea 8: se guardan el valor que ingrese en i.

Linea 9: se reserva memoria para el puntador con malloc vasandose en i

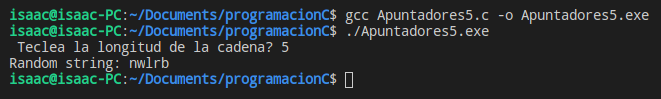
Linea 10: se usa unj if por si no encuentra suficiente memoria regrese un null.

Linea 11: se usa un for para poder datos en la cadena de manera aleatoria.

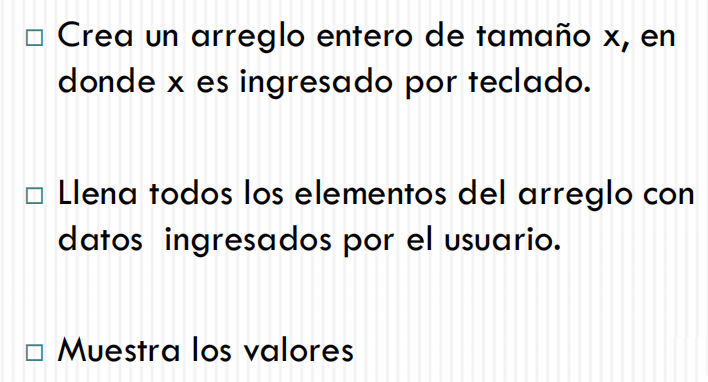
Linea 13: se guarda un cero en la ultima posicion.

Linea 14: se imprime el contenido de buffer.

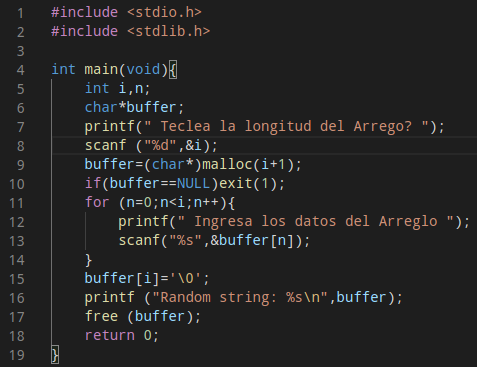
Ejecucion:



Programa 6:



Codigo:



Linea 5: se cra dos variables de tipo entero.

Linea 6: se crea un apuntador.

Linea 7: se manad un mensaje pidiendo la longitud del arreglo.

Linea 8: se guarga el dato en i.

Linea 9: se reserva memoria para buffer.

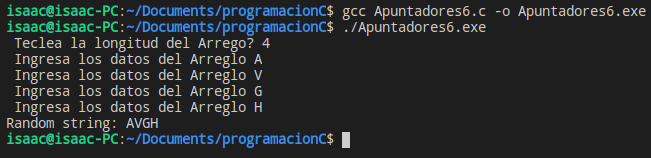
Linea 10: manda null si no encuentra memoria.

Linea 11-13: se cra un for para pedir los datros manda un mensaje pidiendo los datos del arreglo y se guardan los datos en buffer en la posicion del for.

Linea 15: se guardan los datos en la ultima posicion.

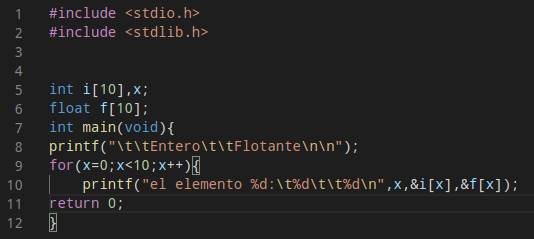
Linea 16: se libera la memoria con free.

Ejecucion:



Programa 7: direciones del Arreglo.

Codigo:



Linea 5: se crea un arreglo de tipo entero con 10 posiciones y una variable.

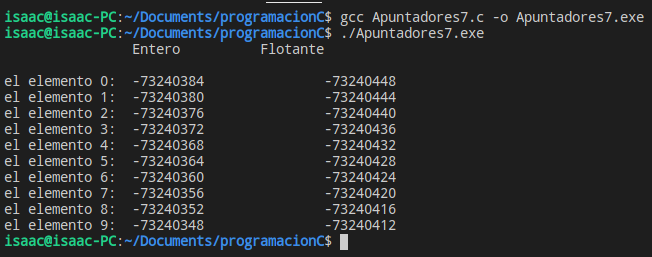
Linea 6: se crea un arreglo de tipo flotante con 10 posiciones.

Linea 8: se imprimen dos letreros entero y flotante.

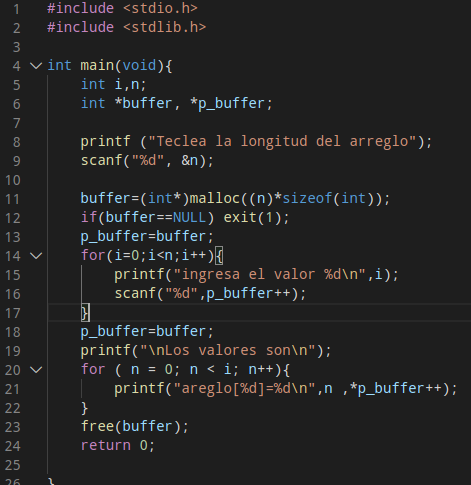
Linea 9: se crea un for para imprimer los valores del arreglo.

Linea 10: se imprimen los valores.

Codigo:



Programa 8:



Linea 5: de tipo entero se crean dos variables.

Linea 6: se crean dos apuntadores.

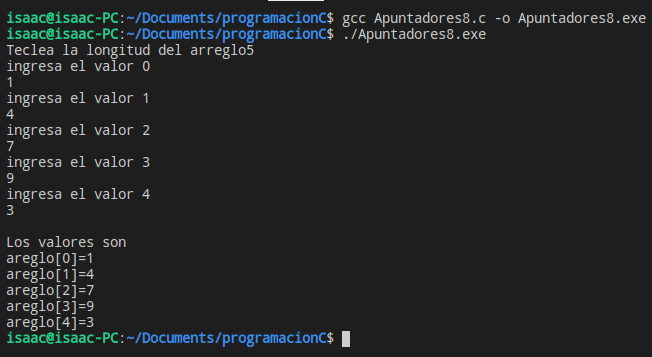
Linea 9-10: se pide la longitud del arreglo.

Linea 11: se reserva la memoria para el areglo.

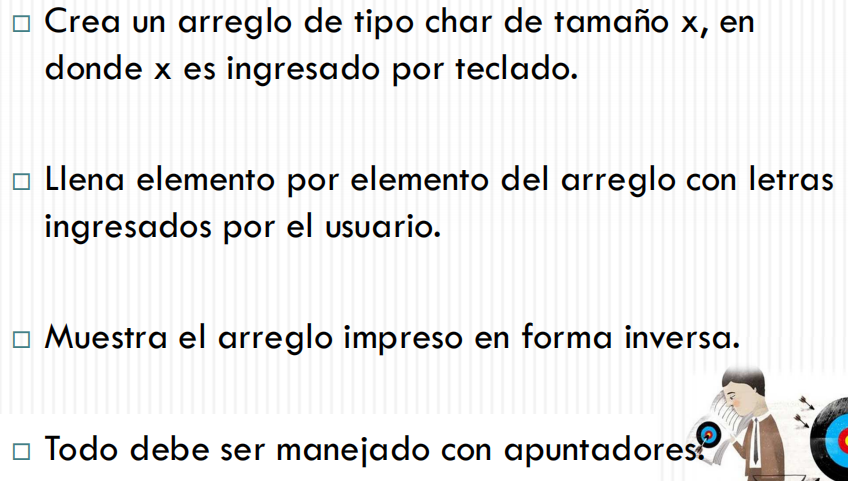
Se imprime los valores.

Y se libera la memoria.

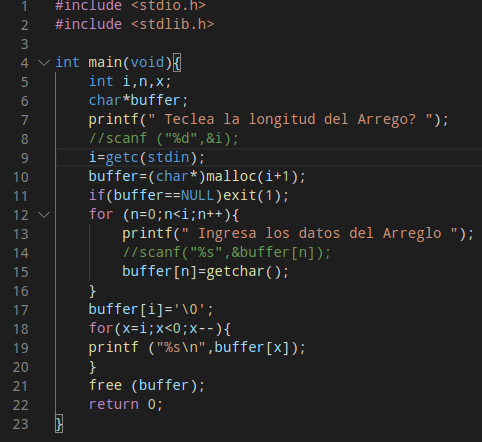
Ejecucion:



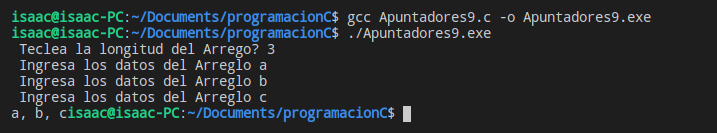
Programa 9:



Codigo:



Ejecucion:



**V. Conclusiones:**

Los apuntadores variablers que podemos decidir el como queremos y en donde guardar las variables para liberar memoria o reservar la memoria que nesecitan para la ejecucion de programas de manera correcta.