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

|  |  |
| --- | --- |
| Profesor(a): | García Morales Karina |
| Asignatura: | Fundamentos de Programación |
| Grupo: | 22 |
| No de Práctica(s): | 9 |
| Integrante(s): | Viveros Fernández Salvador Juan |
|  |  |
|  |  |
|  |  |
| No. de lista o brigada: | 54 |
| Semestre: | Primero |
| Fecha de entrega: | 15/10/2024 |
| Observaciones: |  |
|  |  |

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

Arreglos Unidimensionales.

**Objetivos:**

El alumno utilizará arreglos de una dimensión en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, alineados en un vector o lista.

**Desarrollo:**

**Arreglos:**

Un arreglo son una cadena de datos contiguos que pertenecen al mismo tipo, los cuales tienen un tamaño definido desde el momento en el que son creados, estos datos, además tienen una posición dada dentro del arreglo, los cuales comienzan en la posición 0, culminando en la posición n-1, siendo esta el último valor del arreglo. Así mismo, estos pueden tomar el nombre de vectores.

**Sintaxis del arreglo:**

Para poder declarar, y de ser necesario, inicializar un arreglo, se debe de seguir la estructura de: *(tipo de dato) nombre de la variable [tamaño de la variable] = {datos de la variable};* siendo un ejemplo el siguiente:

int lista [5] = {1, 2, 3, 4, 5};

**¿Qué estructura se requiere para manipular sus índices?:**

Al dentro de un arreglo existir una cadena de datos, cada una con su respectiva posición, si se trata de modificar o imprimir un valor con un printf o scanf, sólo se verá afectado el primer valor de la lista, por lo que se es necesario utilizar una estructura iterativa, siendo posible utilizar las 3 existentes.

**Apuntadores:**

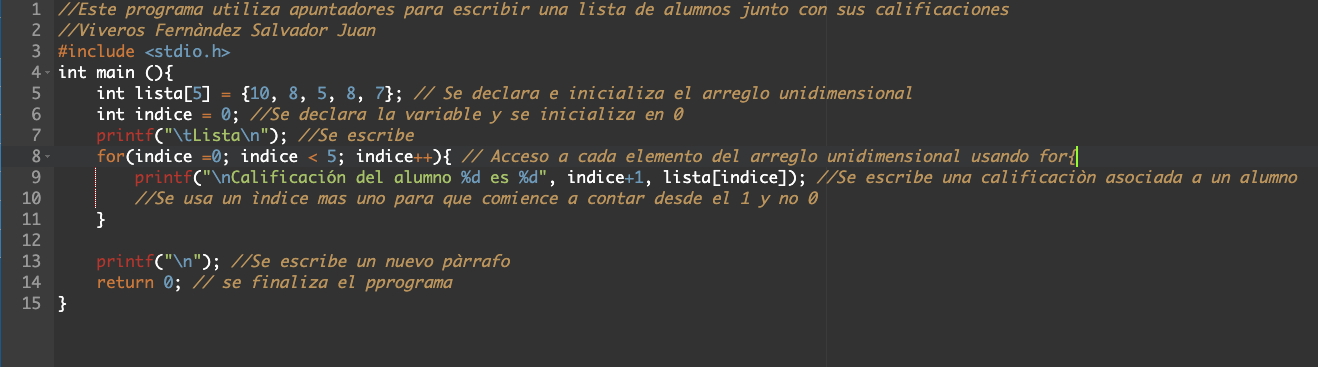
Un apuntador es una variable que al ser inicializada almacena la dirección de memoria de otra variable, así mismo, también es capaz de el valor que la variable contiene, cumpliendo así sus dos funciones. Así mismo, para que una variable y un apuntador se puedan relacionar, estos deben de ser del mismo tipo. Su forma de declararse debe seguir la estructura: *tipo de dato \*nombre del apuntador*, siendo un ejemplo el siguiente:

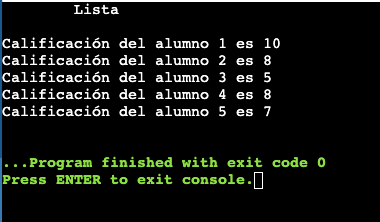
int arreglo[], \*apArr;

apArr = &arr;

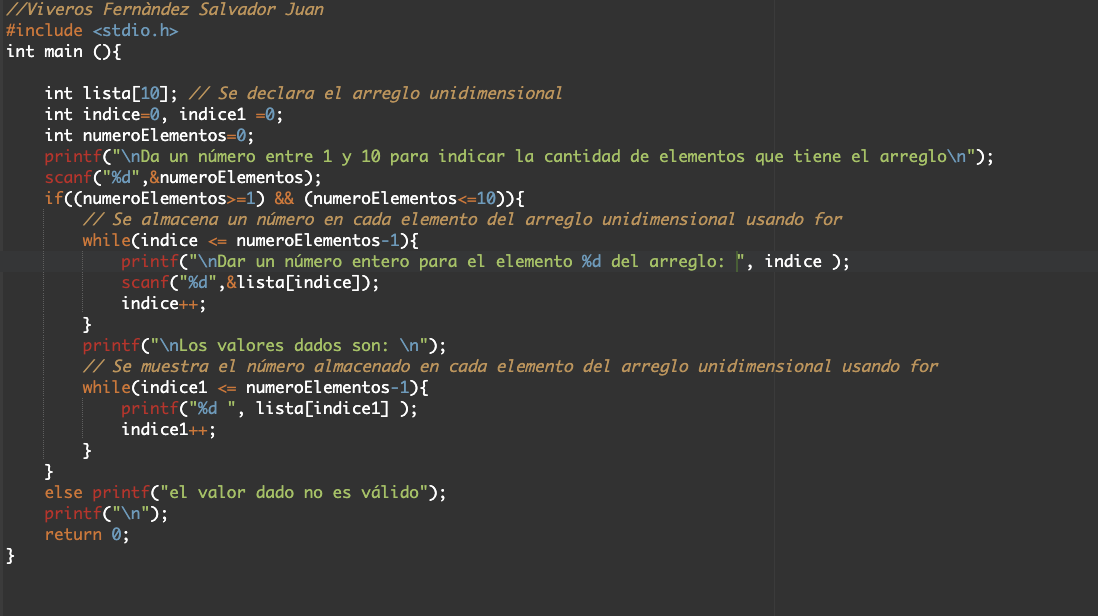
**Ejercicios:**

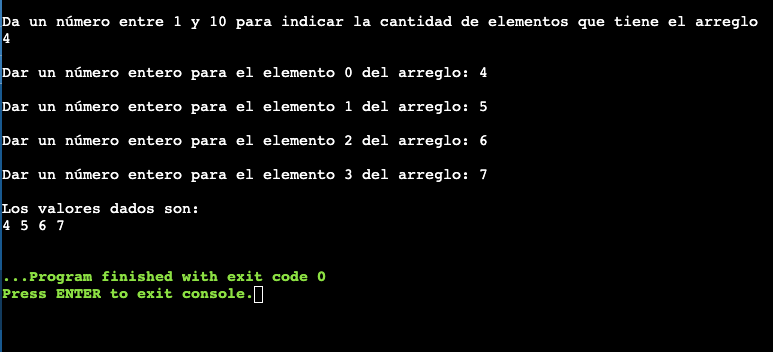
1. Se crea un programa que usando la estructura for, imprime la lista de calificaciones de 5 alumnos, estando esas calificaciones previamente proporcionadas:



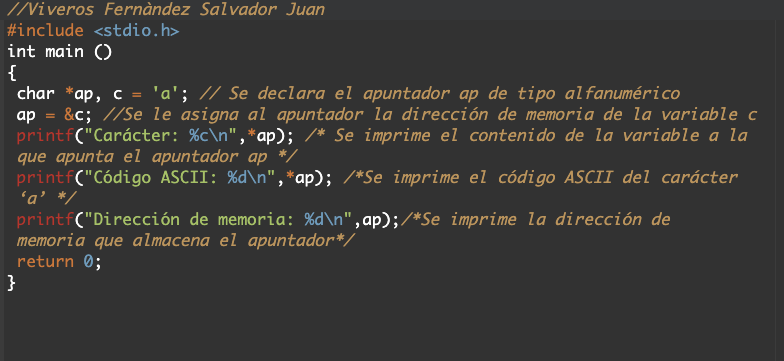


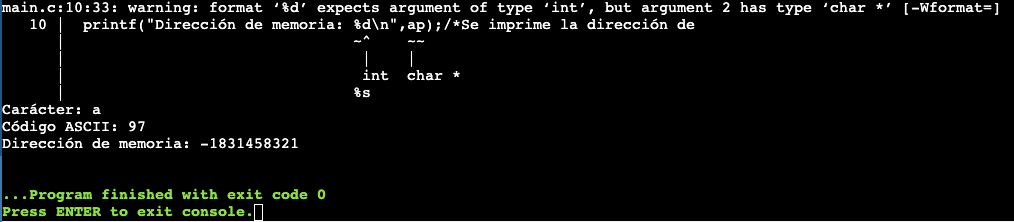
1. Se codifica un programa que le solicita la cantidad de caracteres al usuario, para posteriormente solicitar los mismos caracteres y, finalmente, imprimirlos.



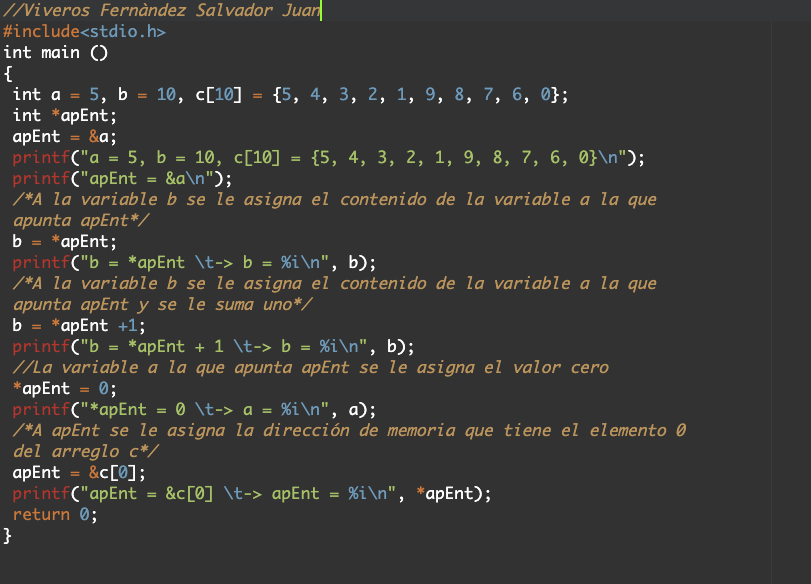


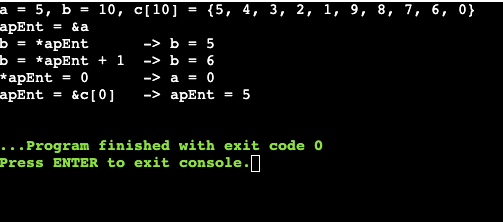
1. Se crea un código que, dada una letra, escribe esa misma letra, su código ASCII, además de, usando un apuntador, su dirección de memoria:



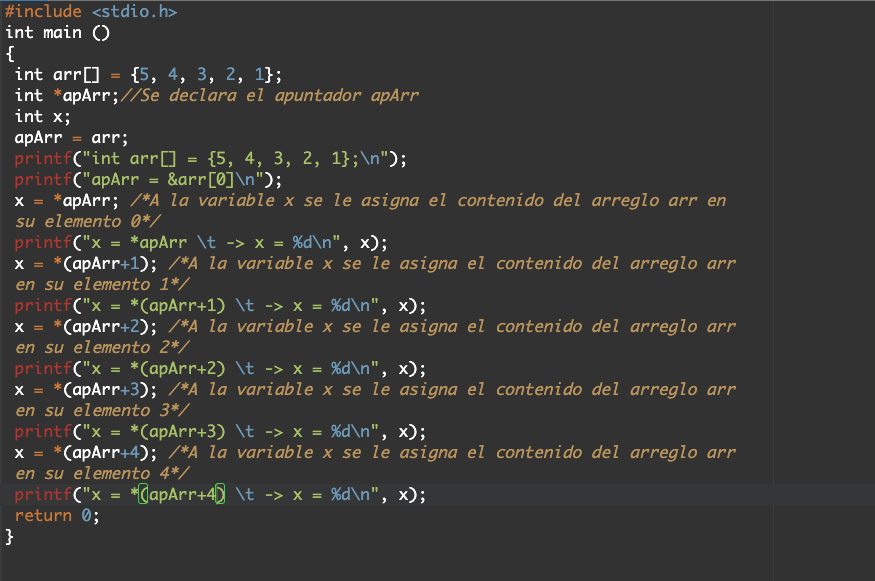


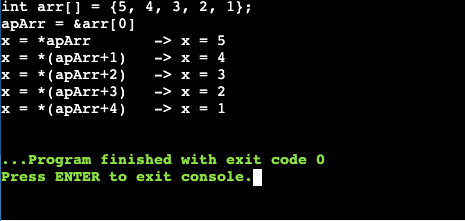
1. Utilizando apuntadores y variables con valores ya definidos, se crea un programa en el que el apuntador toma el valor de la variable a y, modificando el valor de esa misma, el apuntador toma diferentes valores:



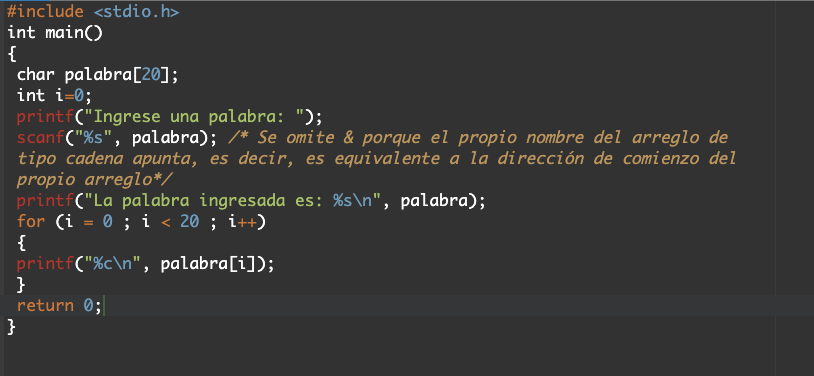


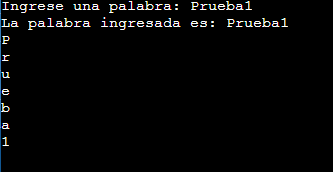
1. Se crea un programa, el cual muestra, utilizando un apuntador, imprime manualmente cada valor de un arreglo de tipo entero:





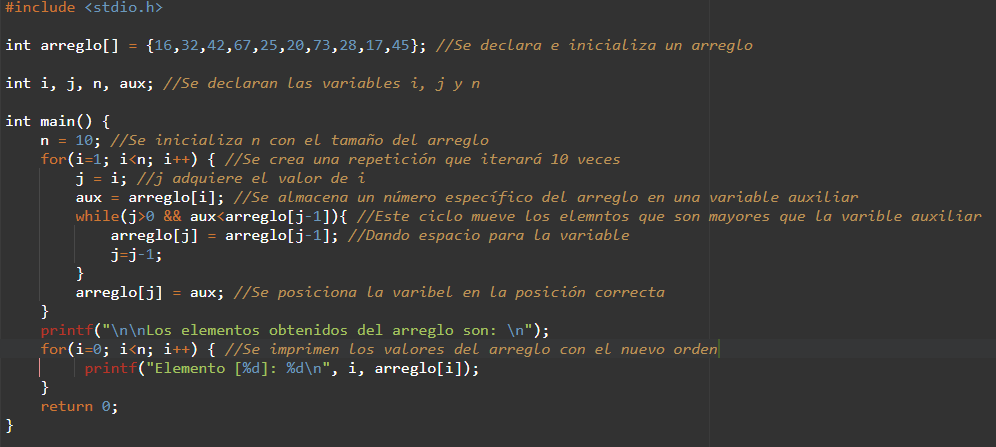
1. El programa solicita que se escriba una palabra, la cual se almacena en un arreglo, posteriormente, usando un ciclo for, se imprime letra por letra de dicha palabra:

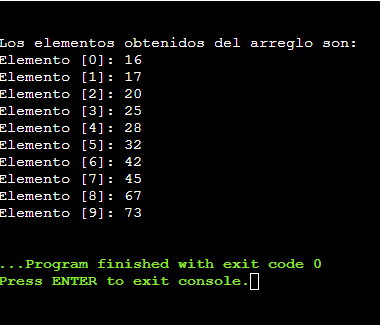




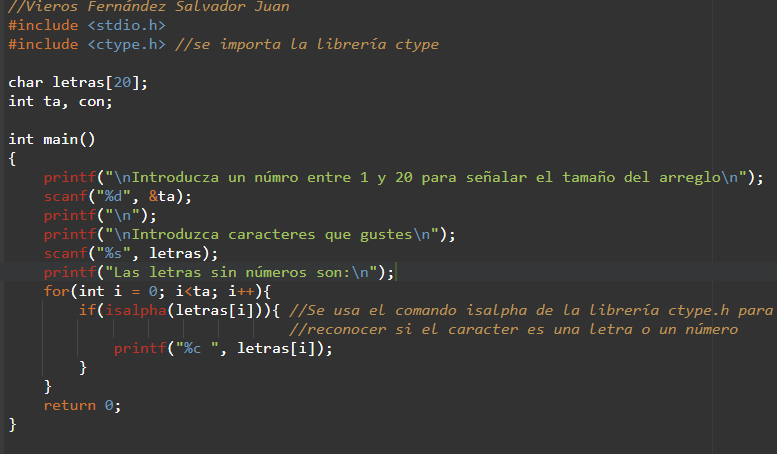
**Tarea:**

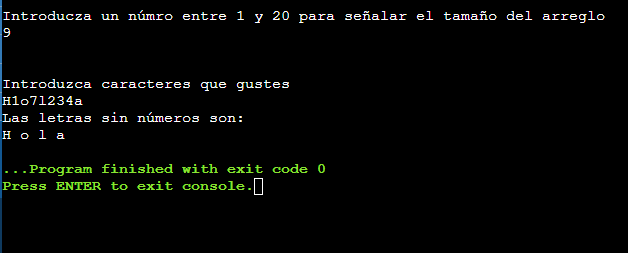
1.-Indica que realiza el siguiente programa: El programa acomoda los valores dentro de un arreglo en orden de menor a mayor.



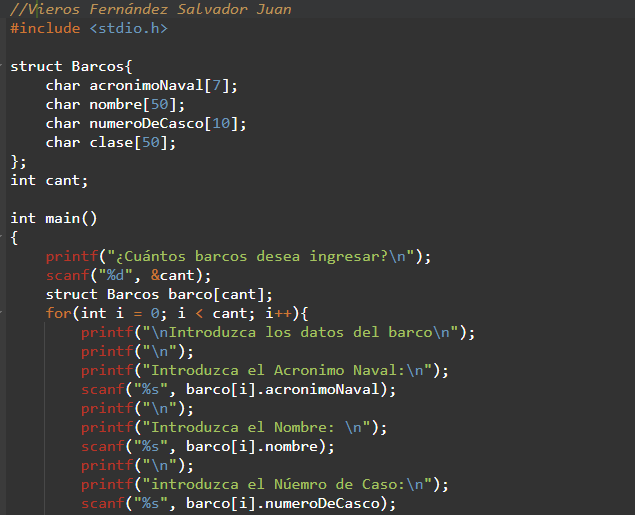


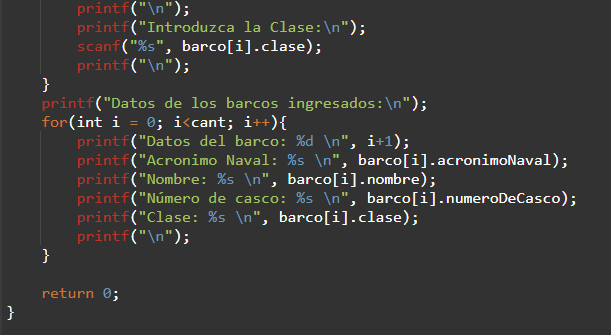
2.- Genera un programa que lea solicite una cadena de letras y números al usuario (emplear arreglo) e imprima solo letras.

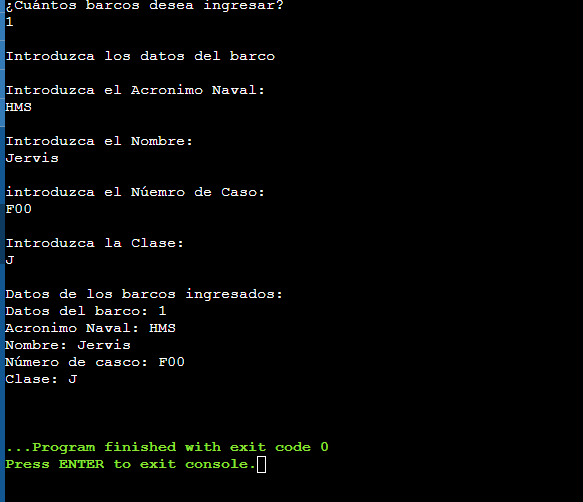




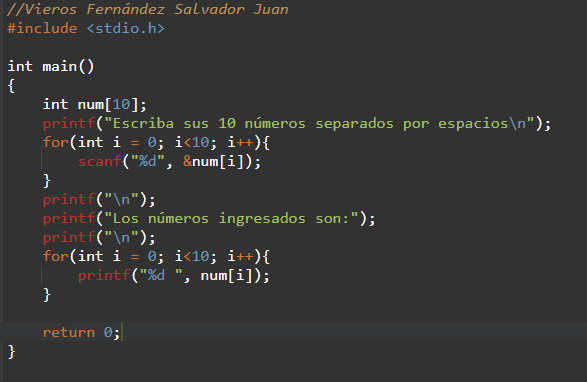
3.- Escribe un ejemplo de un arreglo tipo estructura

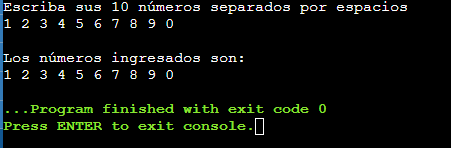




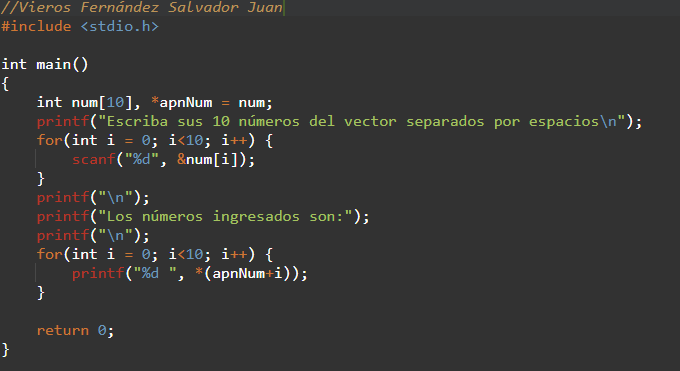


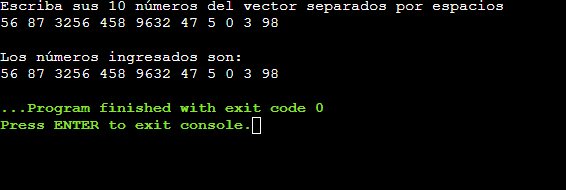
4.- Genera un programa que solicite al usuario un vector de 10 enteros haciendo uso de un arreglo y la estructura iterativa para recorrer por medio de sus índices e imprimir en pantalla.





5.- Modifica el programa del ejercicio 4 y aplica el uso de apuntadores.





**Conclusión:**

El lenguaje C permite hacer arreglos y utilizar apuntadores en ellos, estos tienen mucha utilidad en valores enteros, puesto a que permite almacenar distintos valores en una sola variable, no obstante, en muchos casos, personalmente veo más eficiente utilizar variables de tipo cadena en lugar de arreglos de tipo carácter.

**Bibliografía:**

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