

“Practica 9”

Arreglos unidimensionales

**Objetivo:**

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 es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido

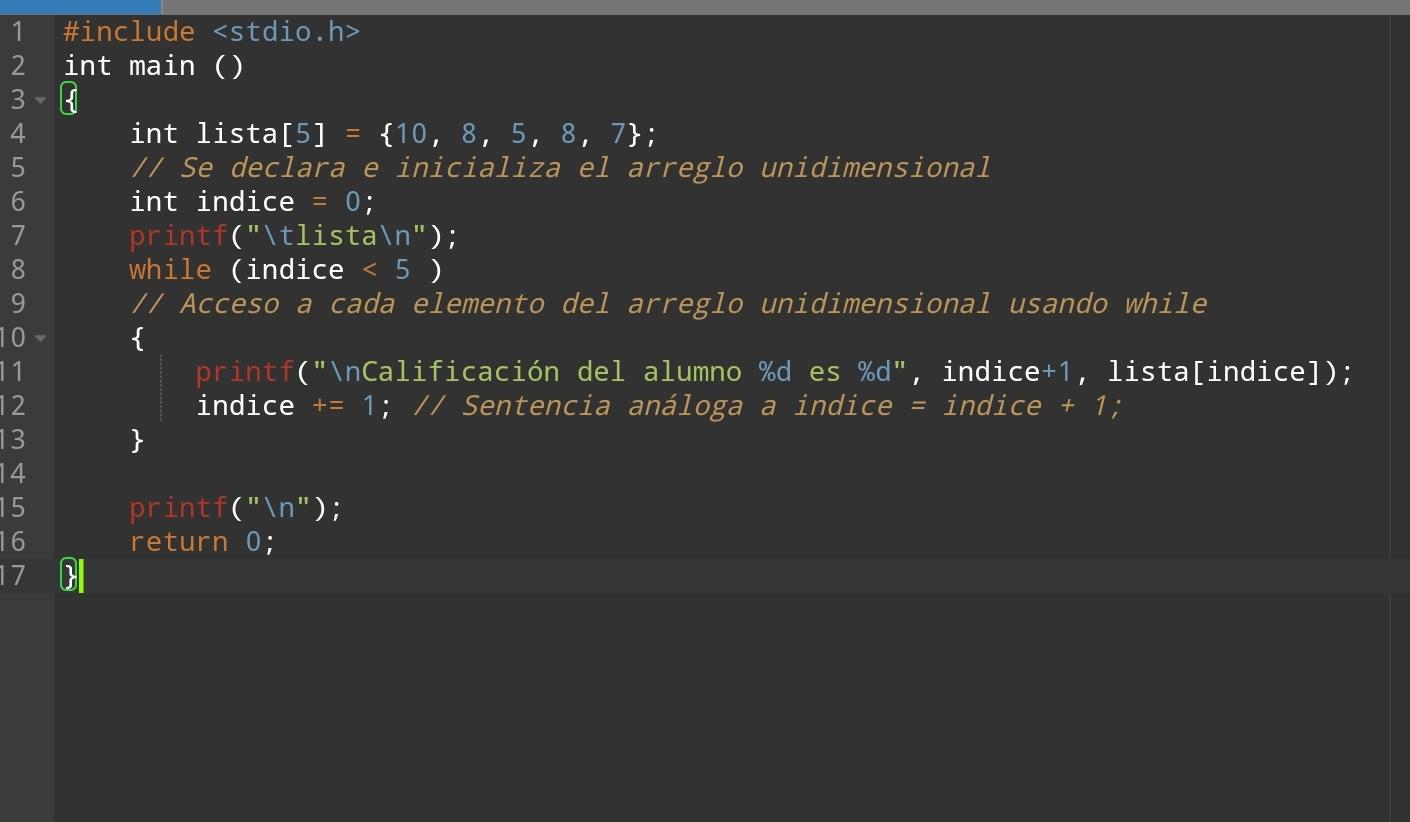
al momento de crearse.

A cada elemento (dato) del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices.

Los arreglos pueden ser unidimensionales o multidimensionales. La dimensión del arreglo va de acuerdo con el número de índices que se requiere emplear para acceder a un elemento del arreglo. Así, si se requiere ubicar a un elemento en un arreglo de una dimensión (unidimensional), se requiere de un índice, para un arreglo de dos dimensiones se requieren dos índices y así sucesivamente. Los arreglos se utilizan para hacer más eficiente el código de un programa, así como manipular datos del mismo tipo

con un significado común.

Dar un ejemplo de arreglo unidimensional:



Sintaxis del arreglo:

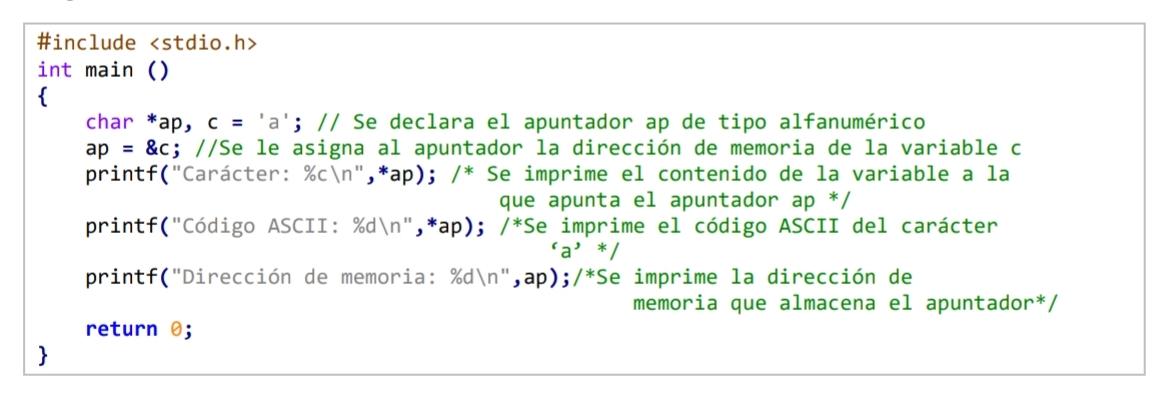
tipoDeDato nombre|tamaño]

¿Qué estructura requieres para manipular un arreglo por medio de sus índices?

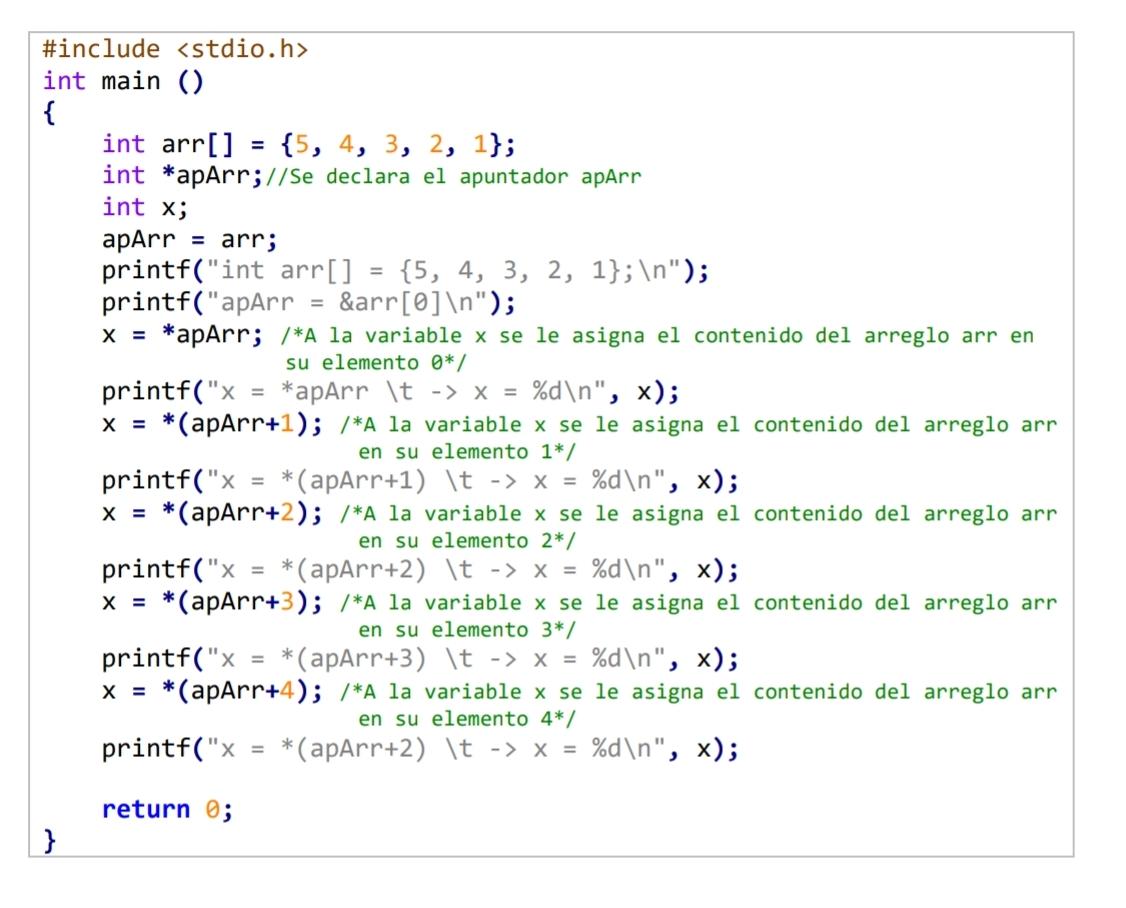
La estructura principal que se necesita es un ciclo que use una variable índice que permita acceder a los elementos del arreglo a través de esa variable.

Apuntadores (Dar un ejemplo de declaración y asignación):

Declaración=

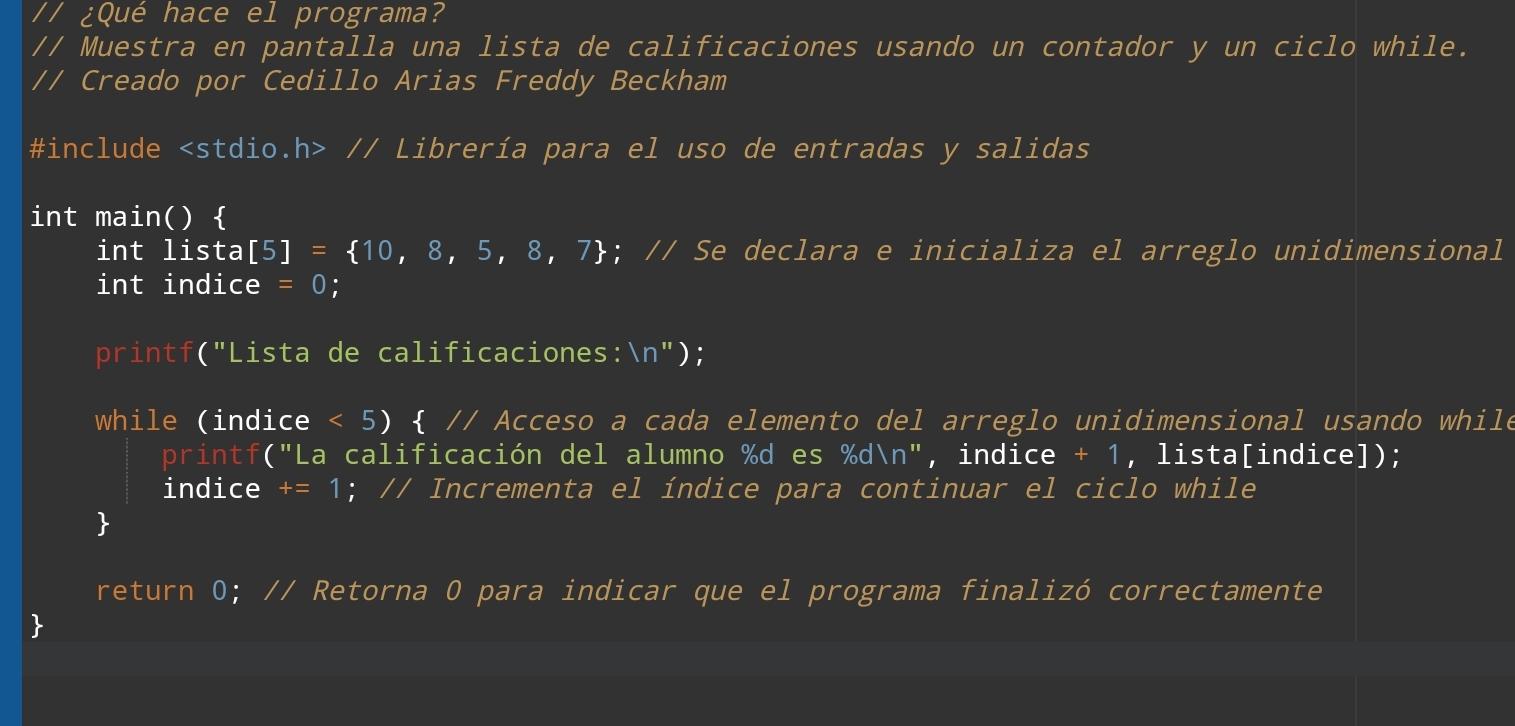


Asignación =

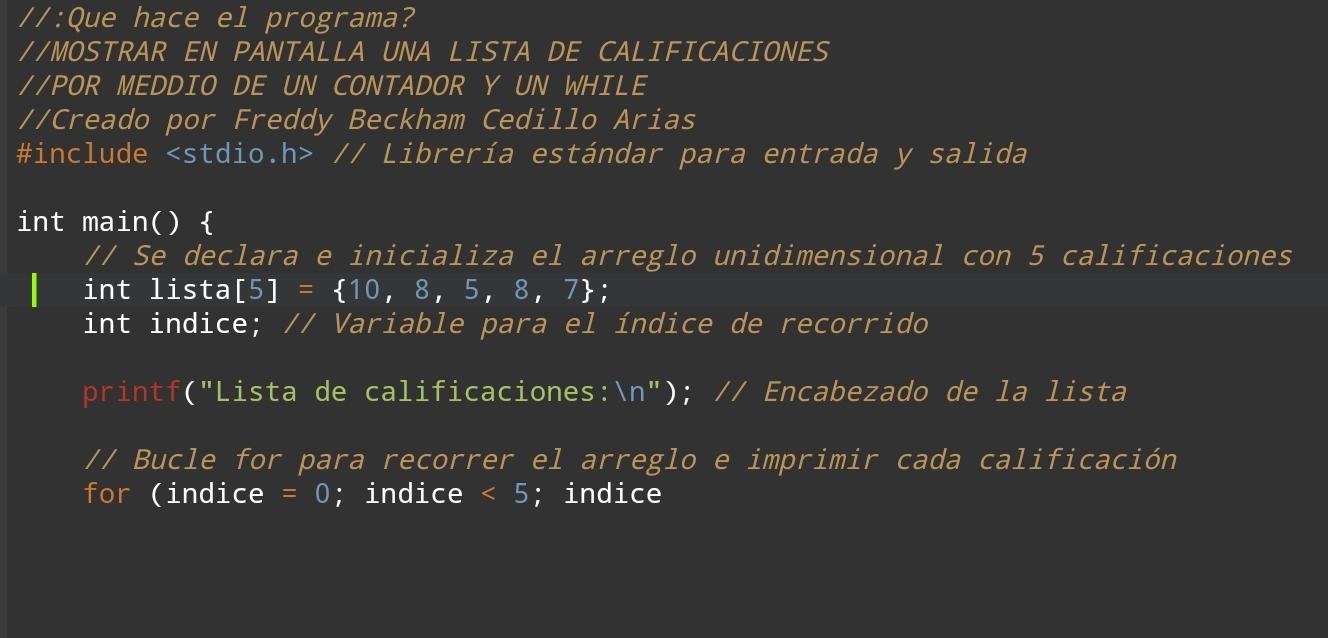


Durante la práctica se hicieron los siguientes códigos:

-Este programa crea un arreglo con 5 elementos. Para acceder, recorrer y mostrar cada elemento, se usa una variable índice que va de 0 a 4, utilizando un ciclo while.



-Se presenta un código que crea un arreglo de 5 elementos. Para poder recorrer y mostrar cada uno de estos elementos, se utiliza una variable llamada índice, que va desde 0 hasta 4, dentro de un ciclo for. El código demuestra cómo funciona este proceso. Sin embargo, en lugar de usar for, el programa usa un ciclo while para hacer lo mismo.



**Tarea:**

1.-Indica que realiza el siguiente programa:

#include <stdio.h>

int arreglo[] = {16,32,42,67,25,20,73,28,17,45};

int i, j, n, aux;

main() {

n = 10;

for(i=1; i< i++) {

j = i;

aux = arreglo[i];

while(j>0 && aux<arreglo[j-1]){

arreglo[j] = arreglo[j-1];

j=j-1;

}

arreglo[j] = aux;

}

printf("\n\nLos elementos obtenidos del arreglo son: \n");

for(i=0; i< i++) {

printf("Elemento [%d]: %d\n", i, arreglo[i]);

}

return 0;

}

R= Descripción del algoritmo:

El programa implementa el algoritmo de ordenamiento por inserción.

El algoritmo comienza desde el segundo elemento (índice 1) y lo compara con los elementos anteriores para colocarlo en su posición correcta. Este proceso se repite para todos los elementos del arreglo.

Después de la ordenación, el programa imprime los elementos del arreglo en orden ascendente.

2.- Genera un programa que lea solicite una cadena de letras y números al usuario(emplear arreglo) e imprima solo letras. Este arreglo debe ser de caracteres, recuerda que un caracter puede almacenar números pero un arreglo de números no puede almacenar caracteres.

Análisis:

DE: Imprimir solo las letras de un arreglo con una combinación de letras y números.

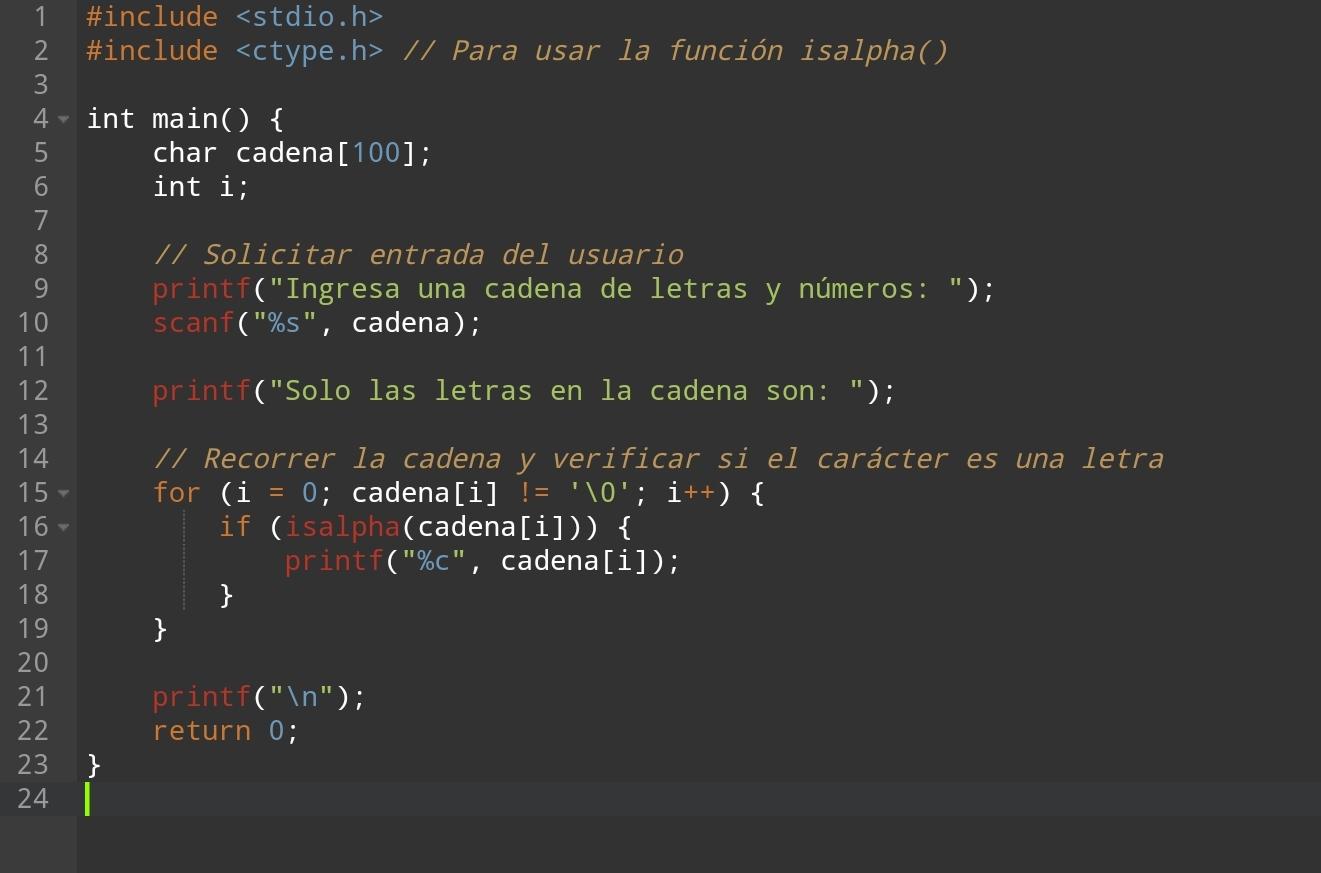
RE: Imprimir solo letras sin números, excluir(0,1,2,3,4,5,6,7,8,9).

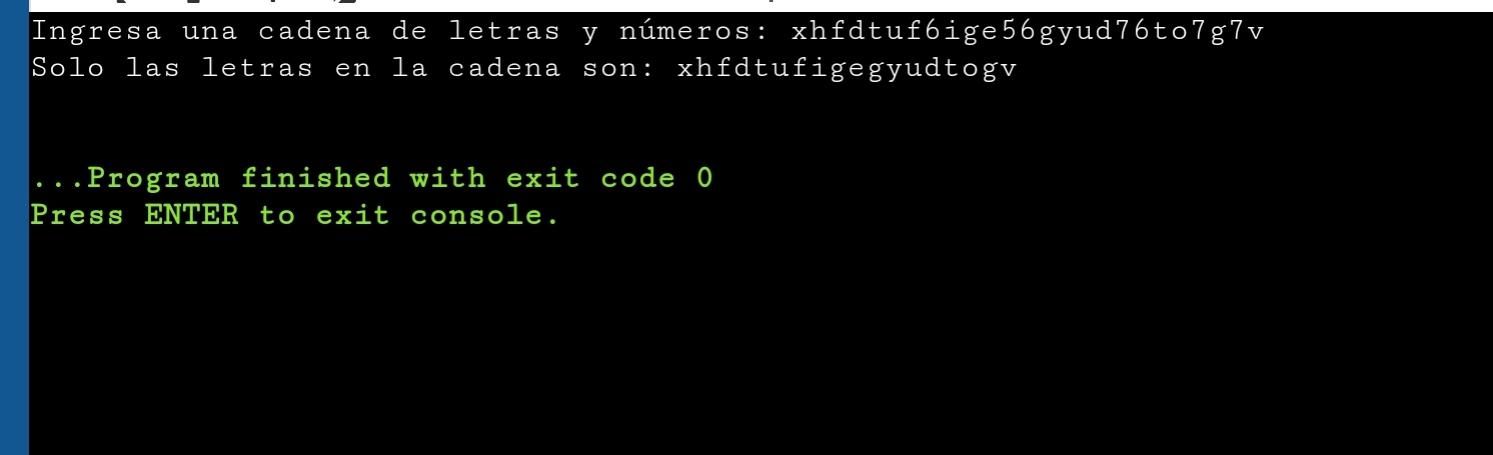
DS: Obtener la impresión de las letras.

Ejemplo:

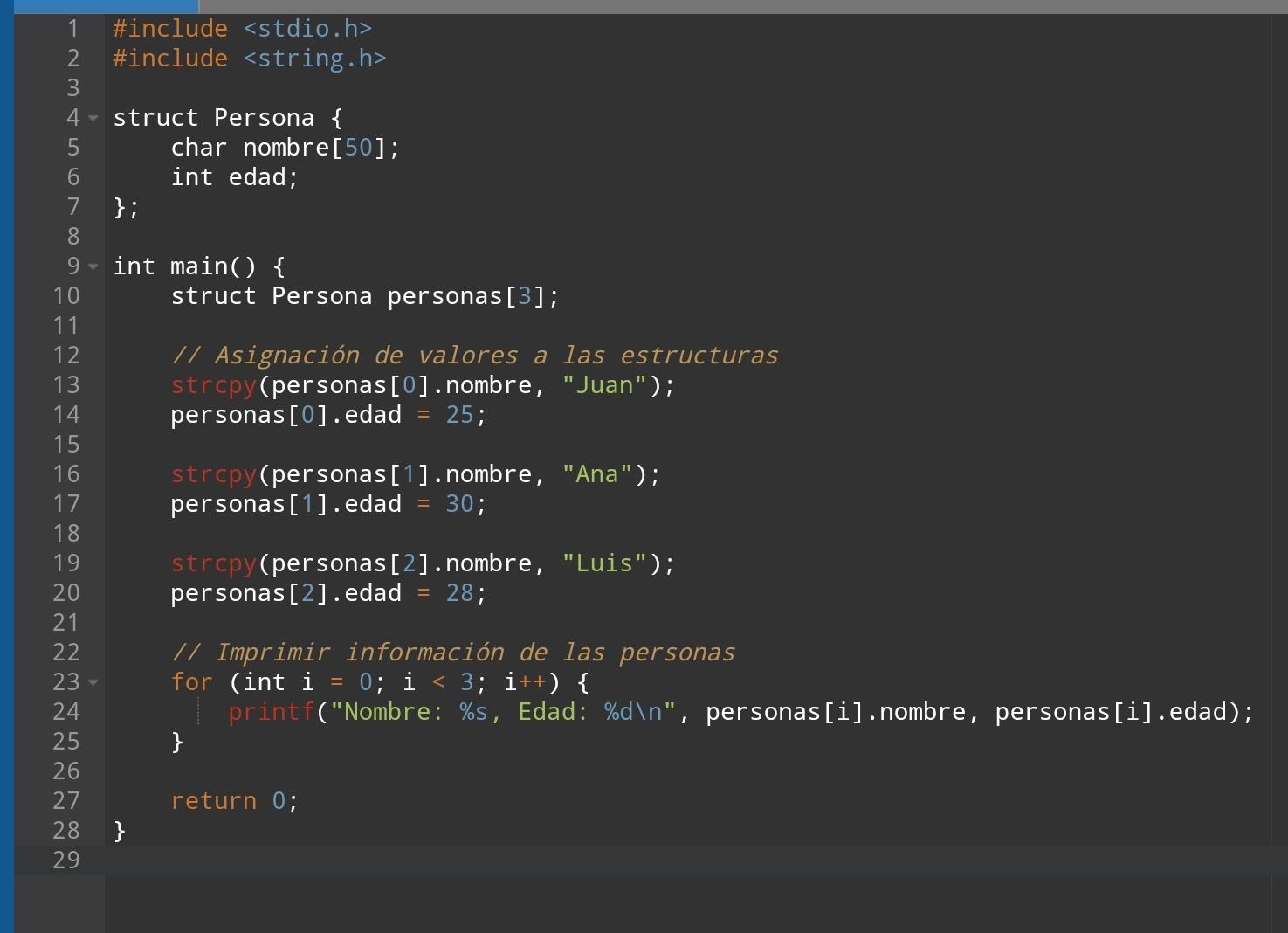
Entrada de usuario= Ej3mpl0c4d3n4

Salida en pantalla= Ejmplcdn



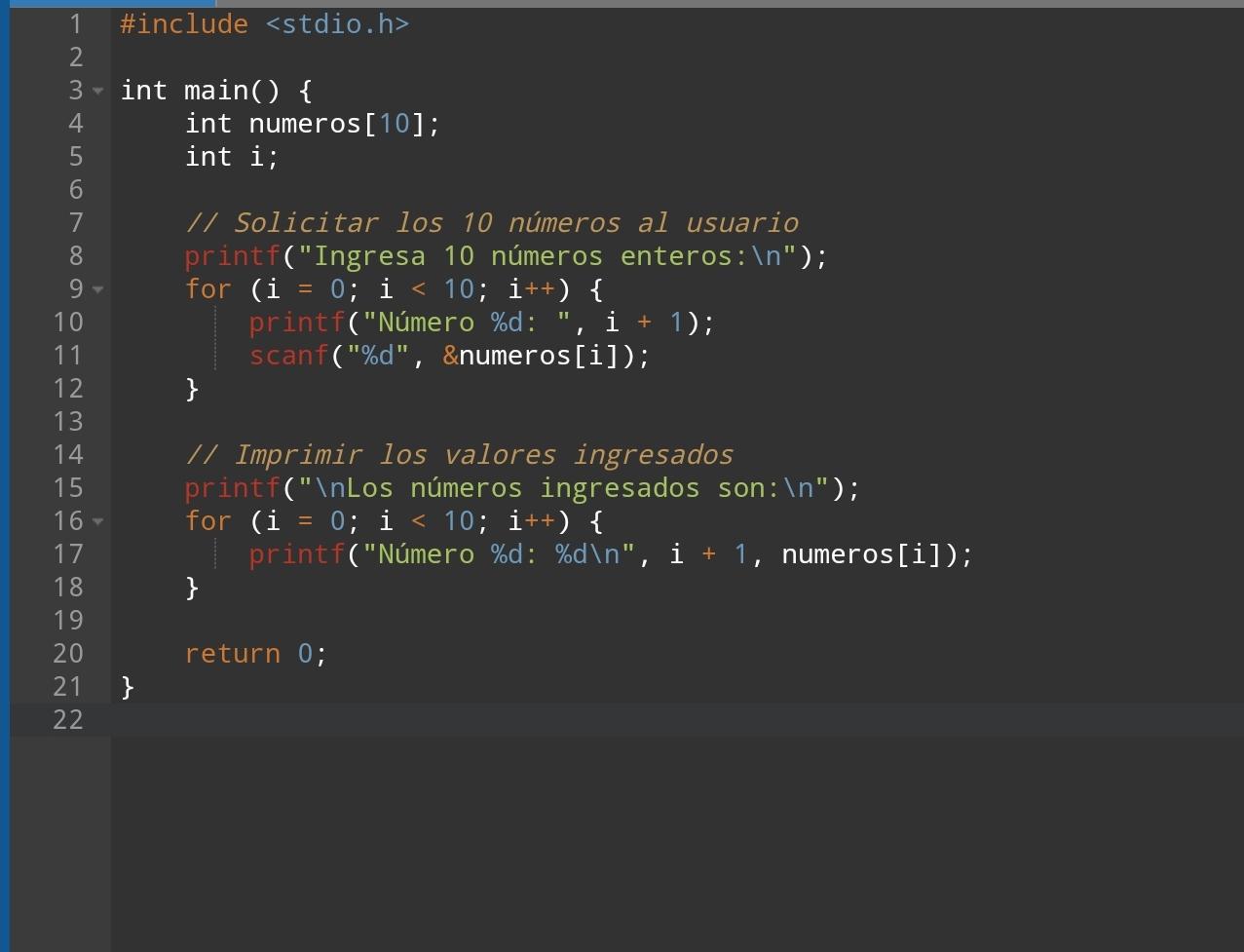


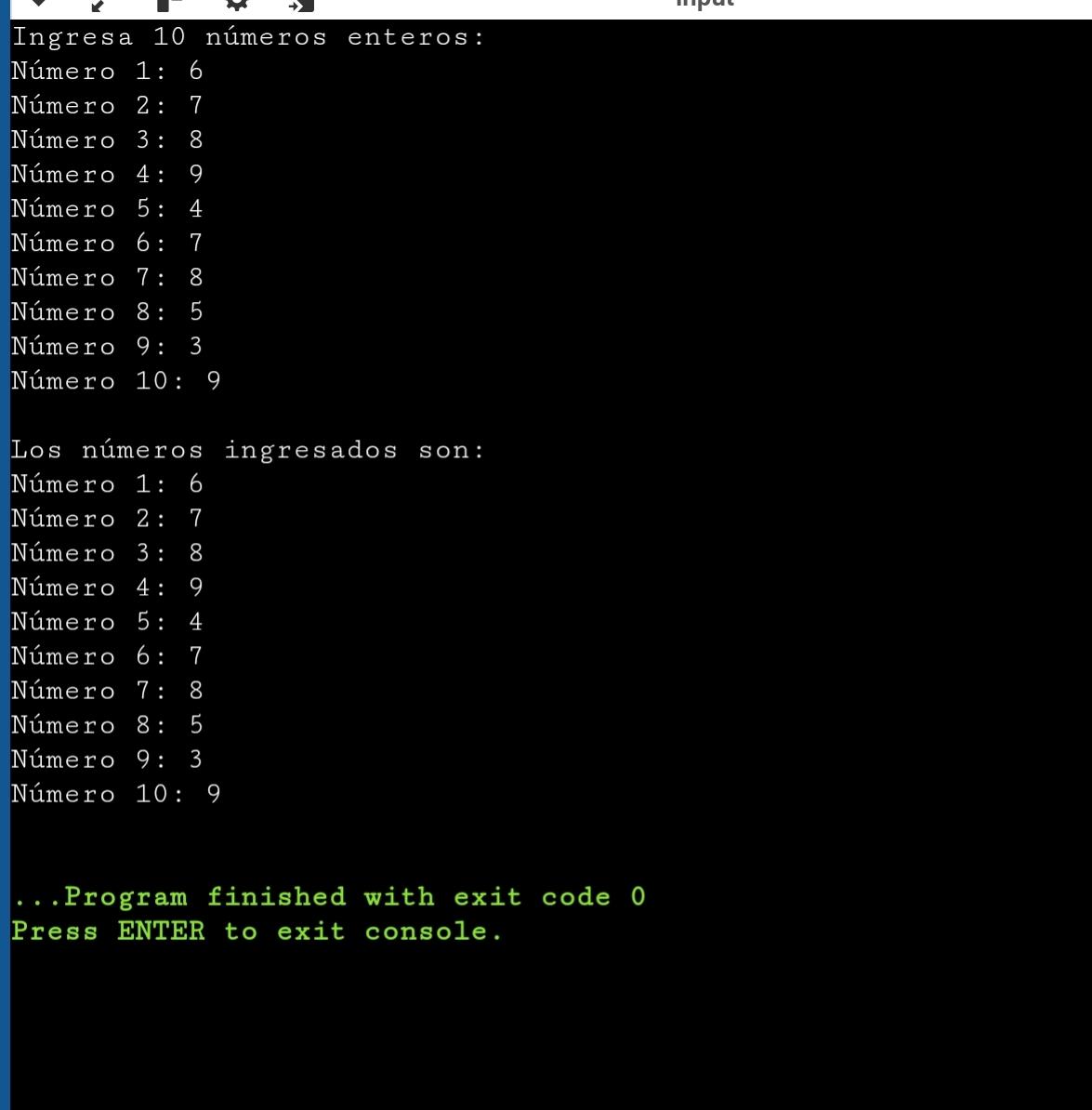
3.- Escribe un ejemplo de un arreglo tipo estructura



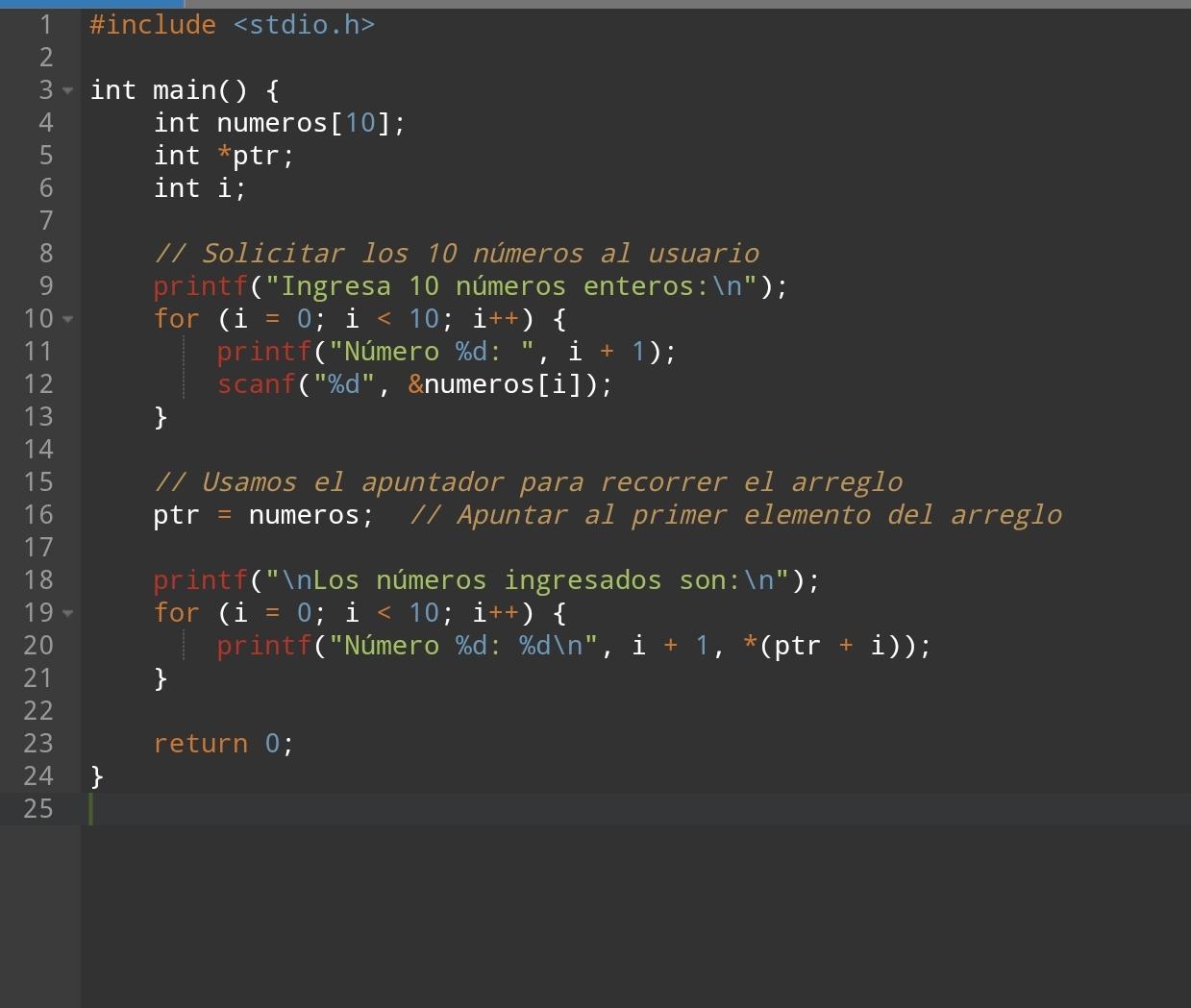


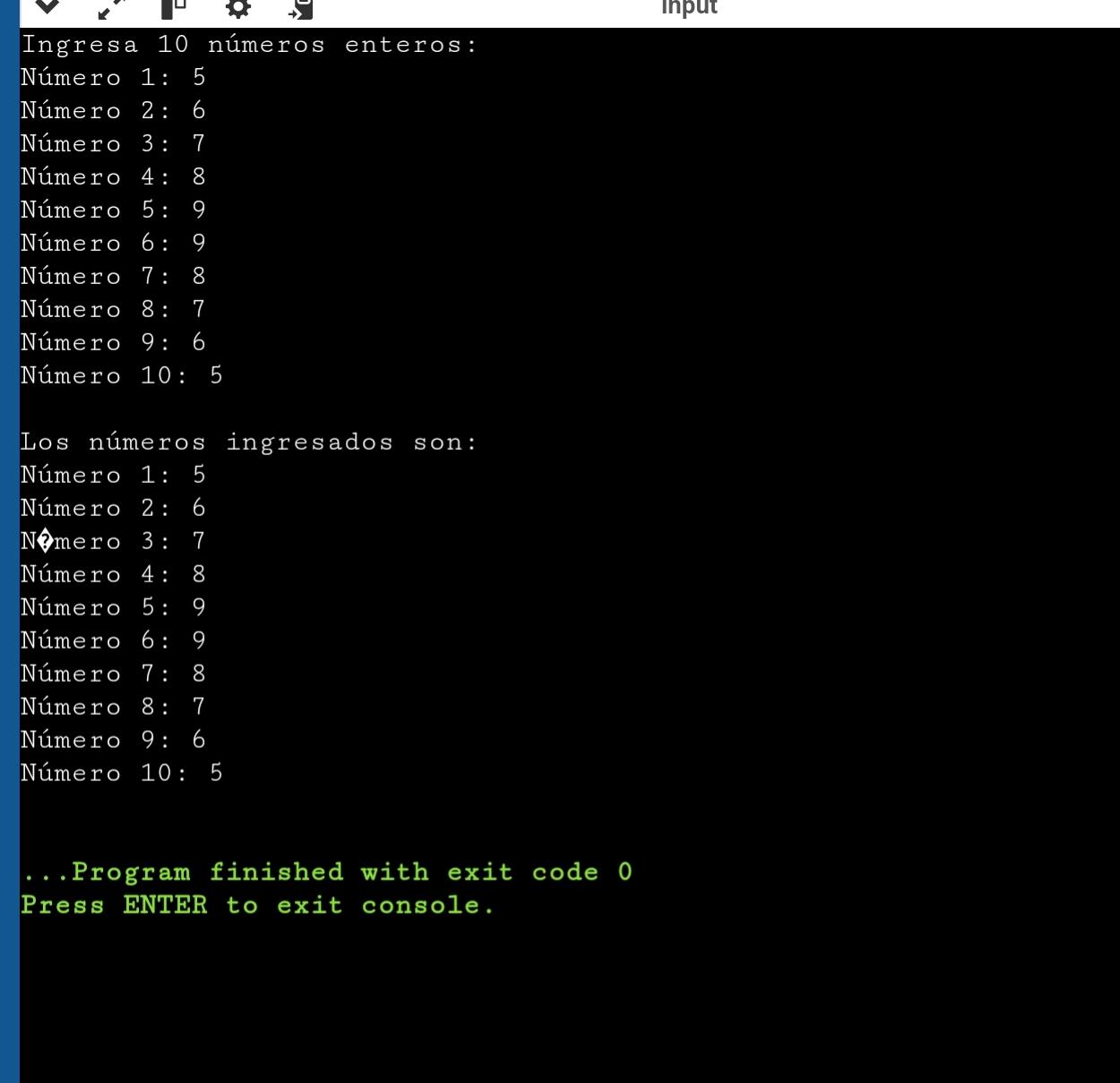
4.- Genera un programa que solicite al usuario un vector de 10 enteros haciendo uso de un arreglo y la estructura iterativa para recorrer por medio de sus índices e imprimir en pantalla.





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





**Conclusión:**

Los apuntadores son variables que nos guardan direcciones de memoria mientras que los arreglos unidimensionales son grupos de elementos organizados en una secuencia, y su nombre actúa como un apuntador al primer elemento. Esto nos permitirá trabajar en un futuro con arreglos de manera eficaz y con mayorfluidez, pero hay que tener cuidado, ya que un uso incorrecto de los apuntadores puede causar grandes errores que podrían ser altamente difíciles de corregir.

**Bibliografía:**

Para esta Práctica no se uso ninguna información web