|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Edgar Tista García |
| *Asignatura:* | Estructuras de Datos y Algoritmos I |
| *Grupo:* | 1 |
| *No. de Práctica(s):* | 2 |
| *Integrante(s):* | Ugalde Velasco Armando |
| *No. de Equipo de cómputo empleado:* |  |
| *No. de Lista:* |  |
| *Semestre:* | 2020-2 |
| *Fecha de entrega:* | de febrero de 2020 |
| *Observaciones:* |  |
|  |  |

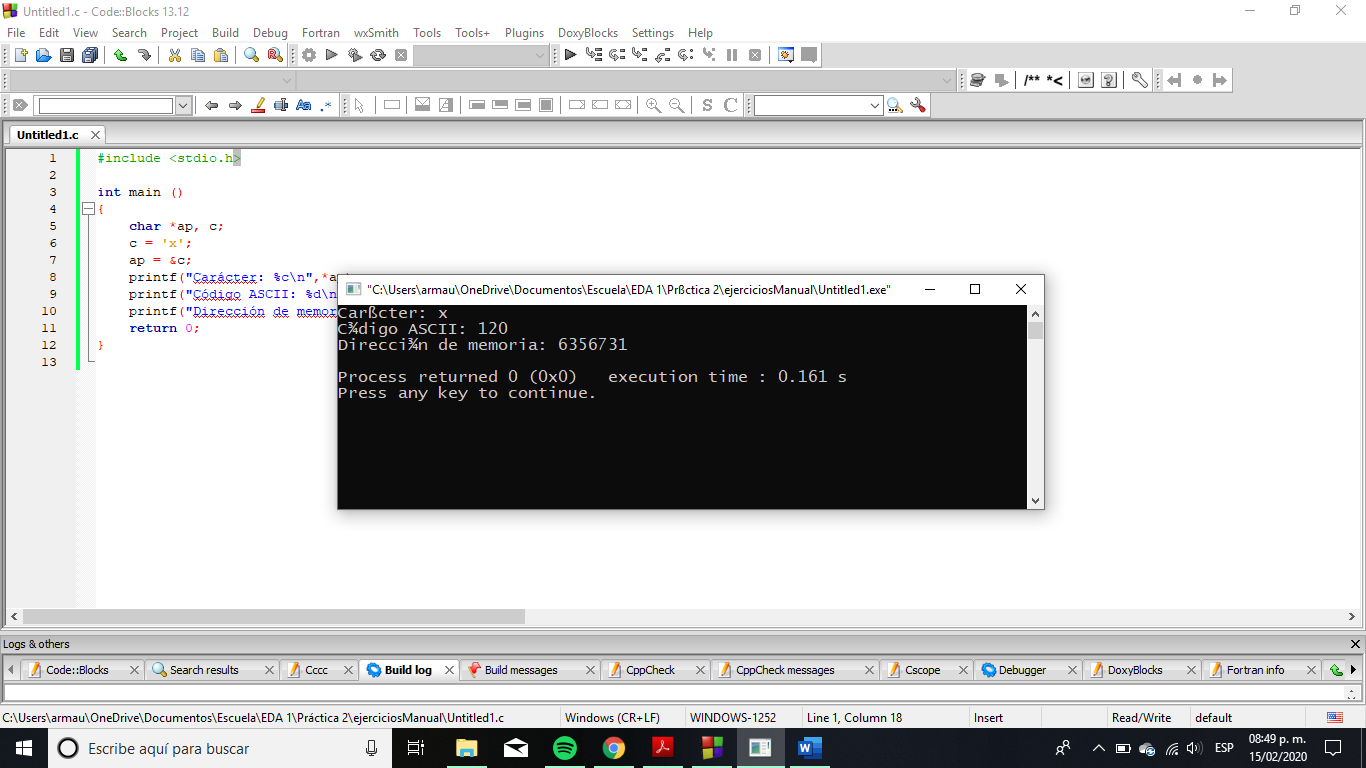
CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Objetivo**

Utilizar apuntadores en lenguaje C para acceder a las localidades de memoria tanto de datos primitivos como de arreglos.

**Ejemplos de la guía de laboratorio**

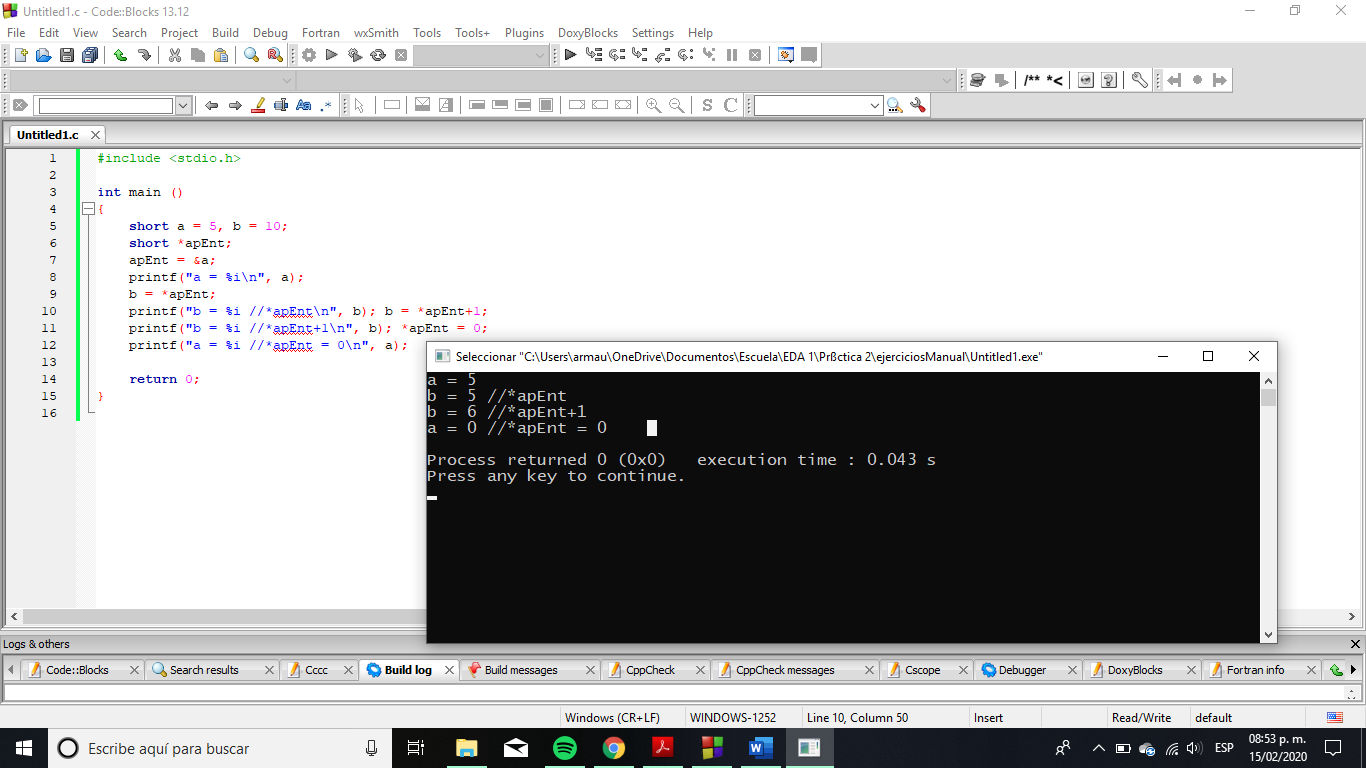
**Apuntador a carácter**



En este programa, primero se imprime el carácter ‘x’, utilizando la función printf y el especificador de formato ‘%c’. Luego, se imprime su valor numérico, utilizando el especificador de formato ‘%d’, el cual es 120. Sabemos que en realidad los caracteres no son más que valores numéricos: en este caso, el estándar de representación de caracteres es ASCII, donde la letra ‘x’ se representa con el número 120.

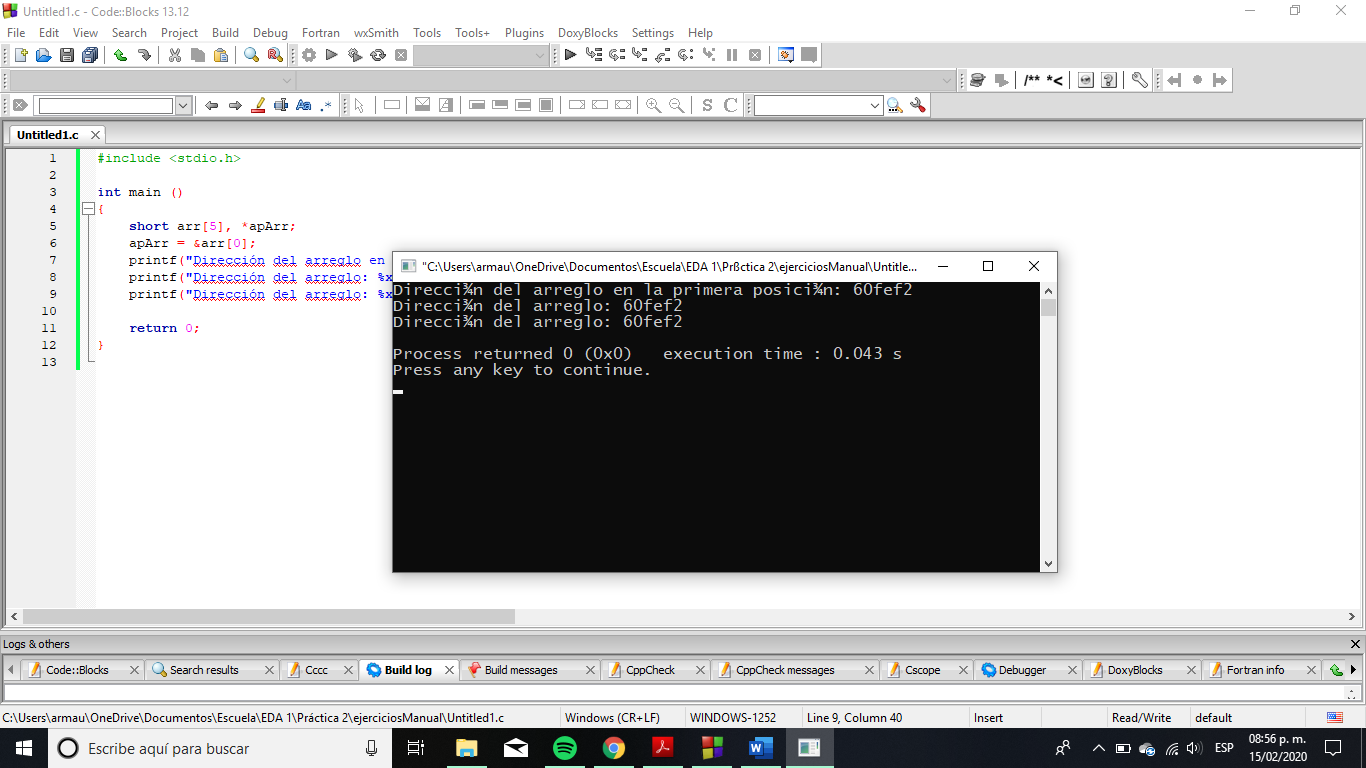
Finalmente, se imprime la dirección de memoria donde el carácter antes mencionado se encuentra almacenado. Esta tarea se lleva a cabo mediante la utilización de un apuntador a la variable que contiene el carácter.

**Apuntador a entero**



En este programa se manipulan las variables a y b, las cuales son de tipo entero. Primero, el apuntador apEnt contiene la dirección de memoria de la variable a. Se imprime el valor de la variable a. Luego, el valor de la variable a la que apunta apEnt, el cual es 5, se le asigna a la variable b y se imprime. Después, se lleva a cabo el mismo procedimiento pero se le suma uno a dicha cantidad. Finalmente, se le asigna el valor de 0 a la variable a la que apunta apEnt, es decir, a la variable a.

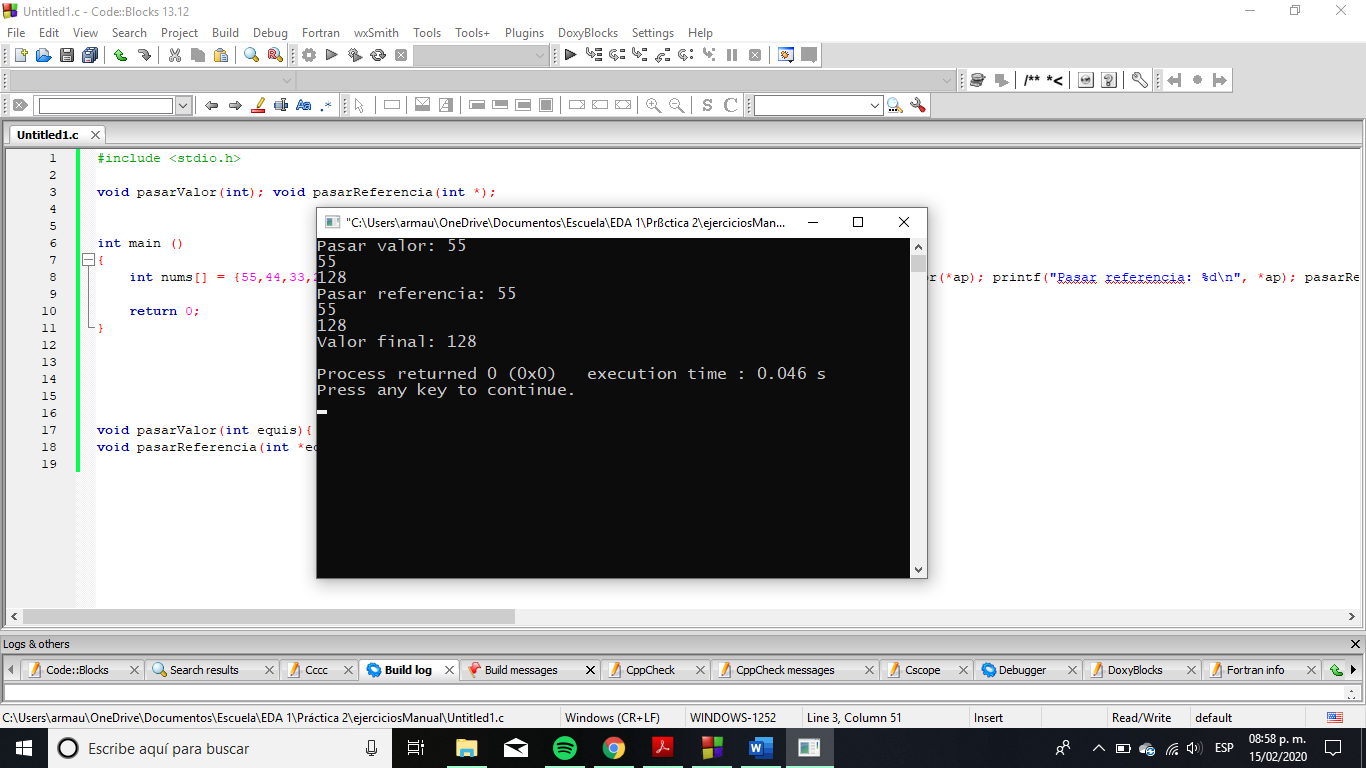
**Apuntador a arreglo**



Este programa demuestra de forma clara la manera en la que los apuntadores a arreglos funcionan. Primero, se imprime la dirección de memoria del primer elemento del arreglo. Posteriormente, se imprime la dirección del arreglo, utilizando el operador de dirección (&) y el identificador de éste. Finalmente, se imprime el valor de un apuntador al primer elemento del arreglo, al igual que en la primera salida.

Podemos percatarnos de que se imprimen los mismos valores en los tres casos, por lo cual podemos confirmar que en realidad un apuntador a un arreglo es equivalente a la dirección de memoria del primer elemento de éste.

**Paso de variables por valor y por referencia**



Este programa ejemplifica claramente el funcionamiento del **paso por valor y por referencia.**

Primero, se imprime el primer valor de un arreglo de enteros anteriormente creado, utilizando el operador de indirección y una referencia al arreglo.

Después, se ejecuta la función ***pasarValor*** con el valor anterior como parámetro. Primero imprime el valor original y posteriormente le asigna el valor 128, imprimiendo el valor final.

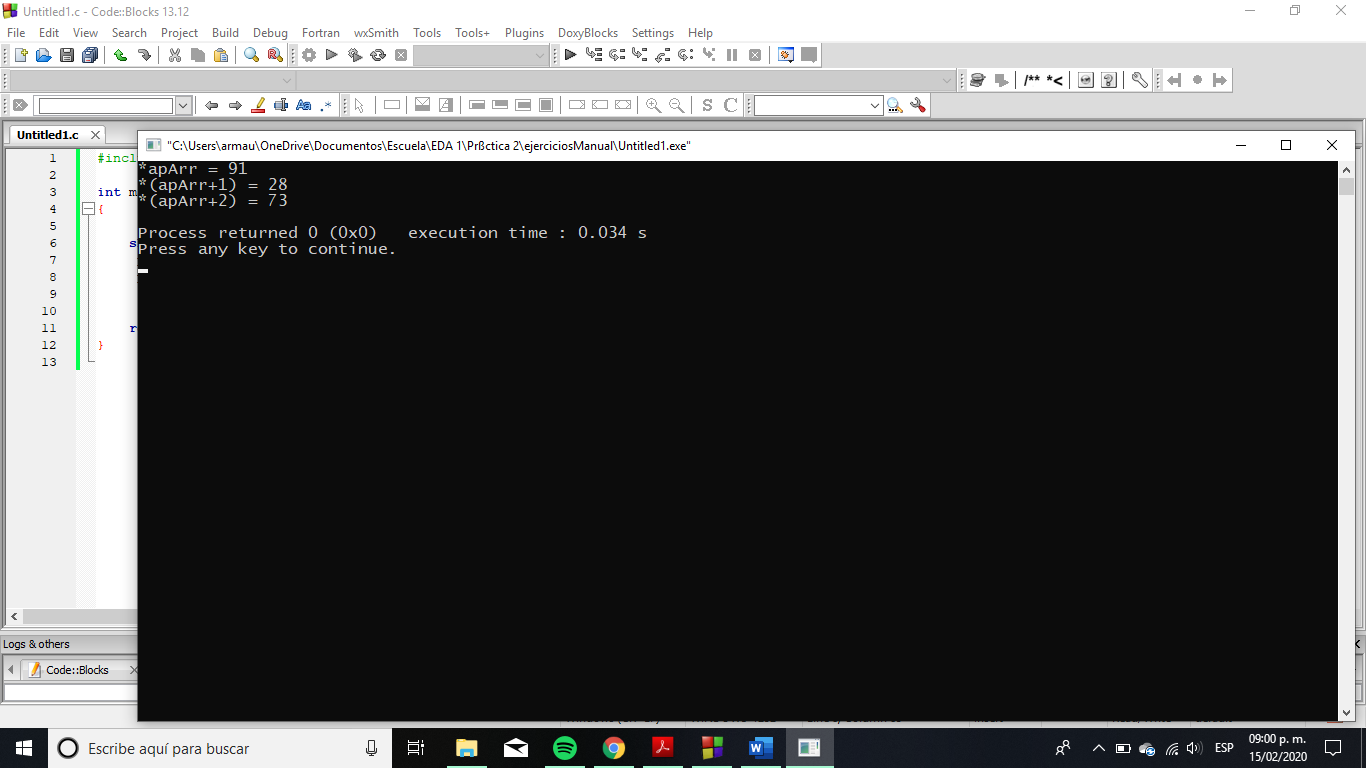
Sin embargo, el valor únicamente es modificado dentro de la función mencionada, ya que claramente se está **pasando por valor** al entero 55.

Después, se imprime el valor actual del primer elemento del arreglo, con el cual se había ejecutado la función anterior. Nos podemos percatar de que, en efecto, el valor permanece igual: es 55.

Posteriormente, se ejecuta la función ***pasarReferencia***, la cual toma como argumento un valor de tipo int\*, es decir, apuntador a entero. Se introduce como parámetro la dirección al primer elemento del arreglo, la cual coincide con el tipo de dato que especifica la función. Al **pasar por referencia**, en este caso el valor en ***main*** sí se modifica a 128.

Finalmente, se imprime el valor del primer elemento del arreglo en ***main***, el cual es 128, como se esperaba.

**Aritmética de direcciones**

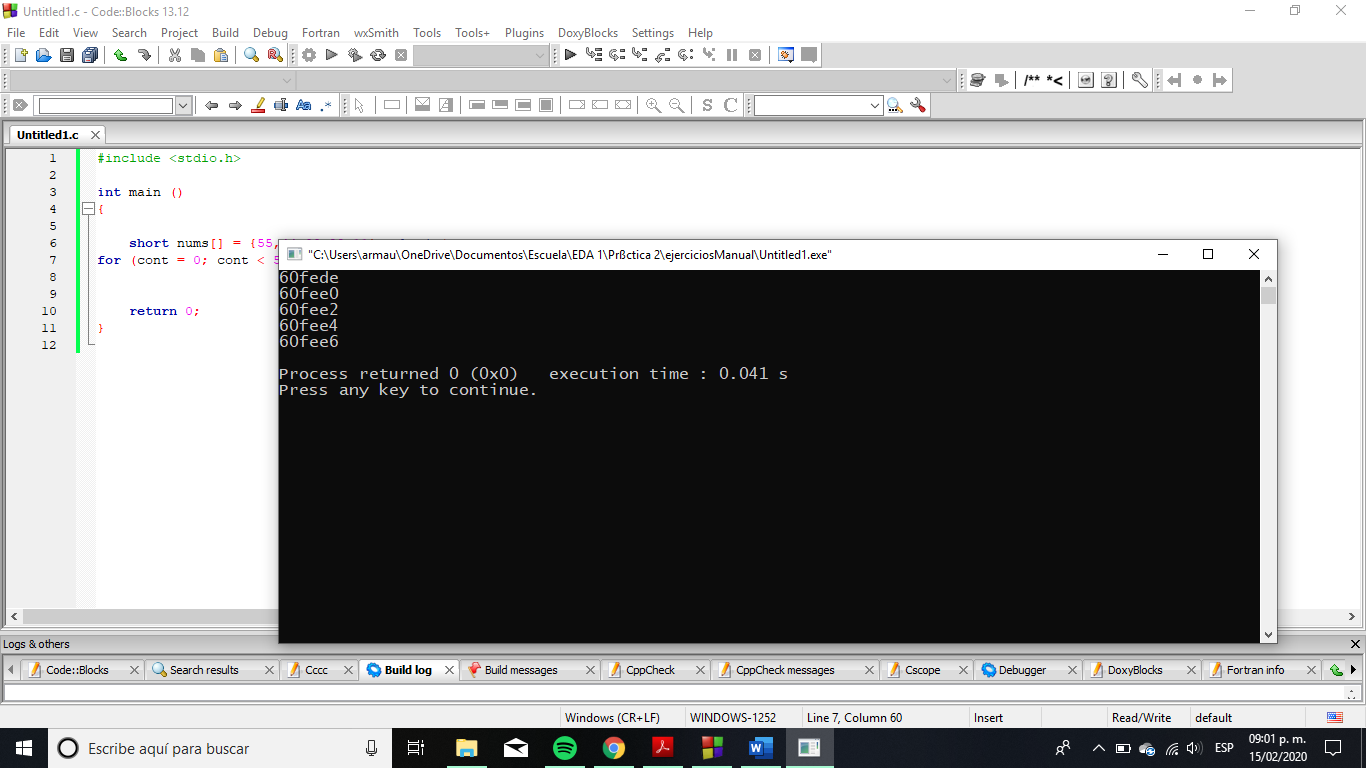


En este programa se imprimen los tres primeros elementos de un arreglo de 5 elementos, utilizando aritmética de punteros.

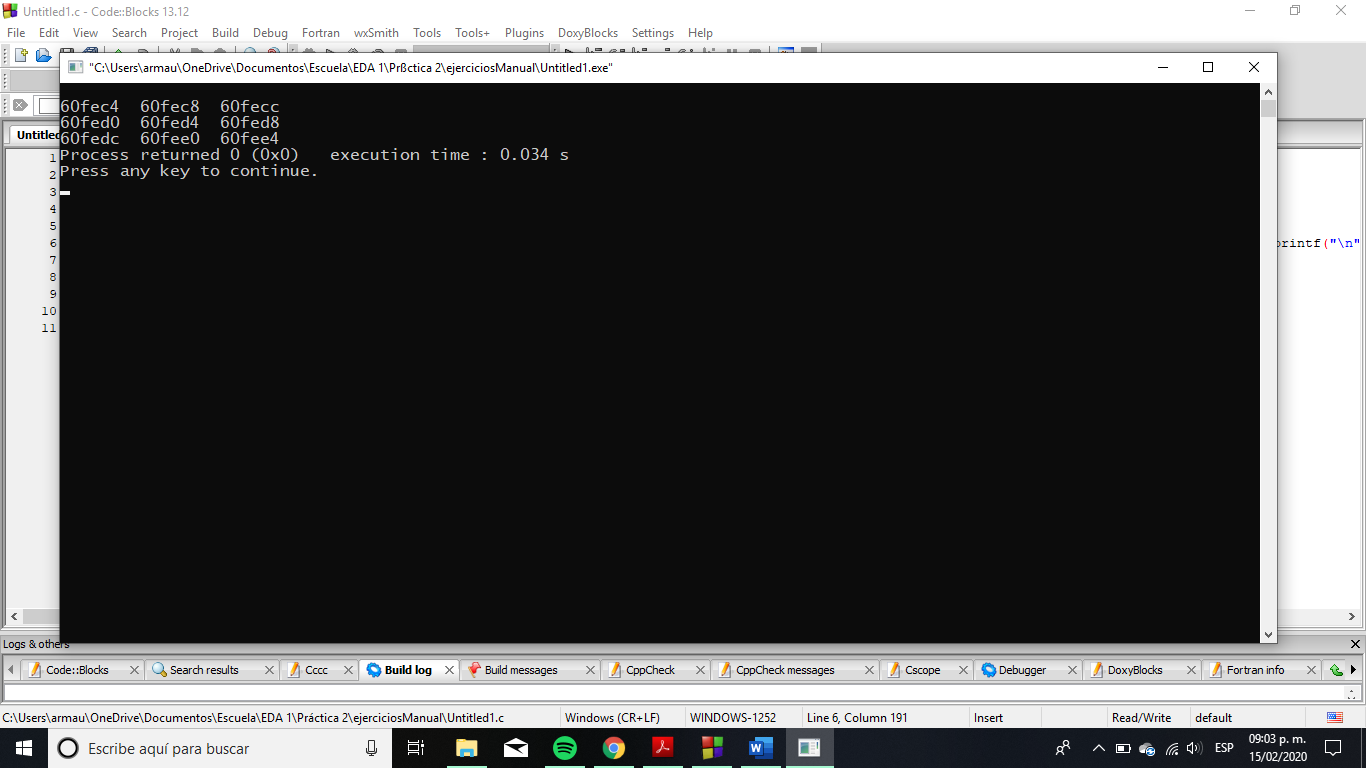
Primero, se crea una variable que contenga un apuntador al arreglo. Sin embargo, sabemos que ésta en realidad es la dirección de memoria del primer elemento, por lo que, al utilizar el operador de indirección, obtendremos como salida el valor del primer elemento del arreglo.

Después, se imprimen dos elementos más del arreglo. El segundo se imprime al sumarle una localidad de memoria a la dirección inicial, y utilizando el operador de indirección, como se muestra en el programa. Finalmente, se imprime el valor del tercer elemento, al realizar la misma operación, pero sumando dos localidades de memoria a la dirección del primer elemento del arreglo.

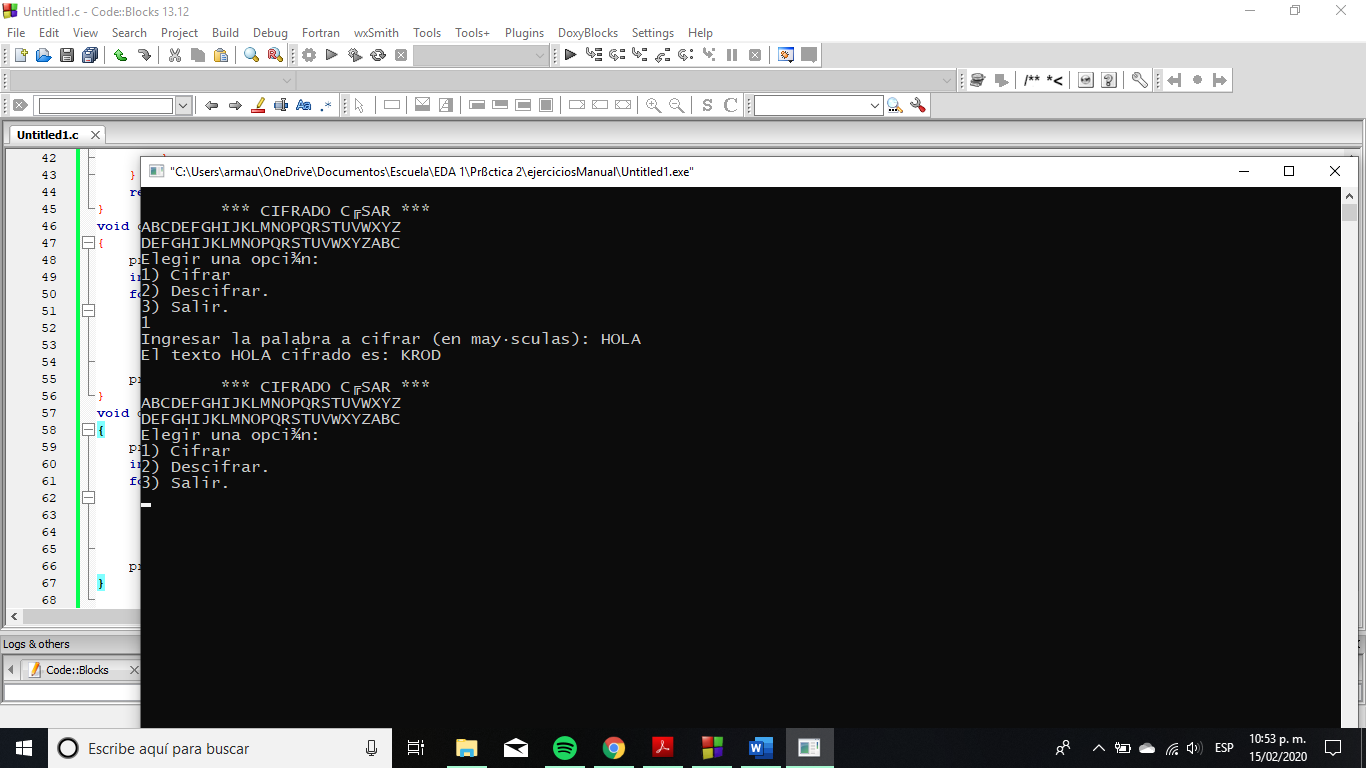
**Arreglo unidimensional**

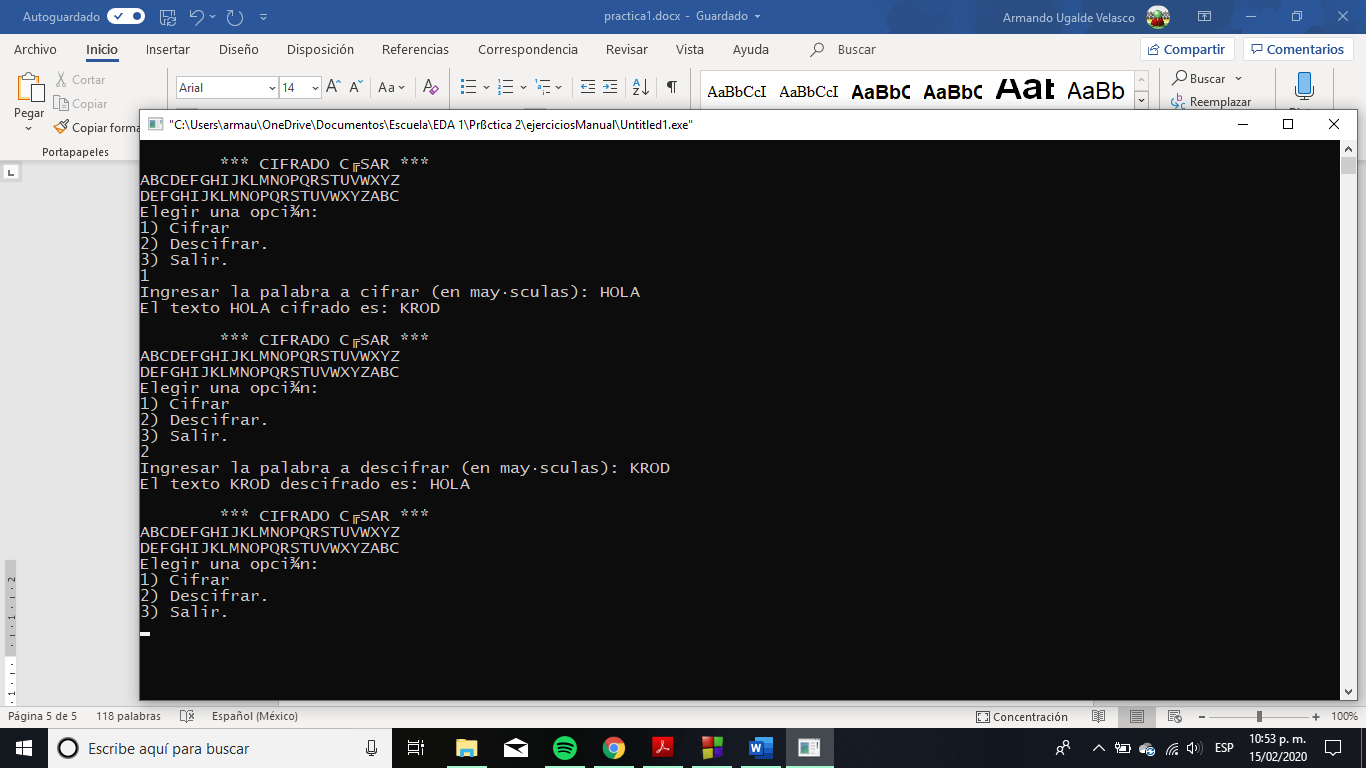


**Arreglo bidimensional**



**Cifrado César**





**Ejercicio 1**

**Conclusiones**

Podemos decir que