**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

Alumno: Vargas Romero Erick Efraín

Profesor: Franco Martínez Edgardo Adrián

Unidad de aprendizaje: Estructuras de datos

Tarea 02: Programación de algoritmos estructurados

Secuencia: 1CM9

Semestre: 2016 – 2017 “A”

Problema 02 3

2.1 Explicación 3

2.2 Algoritmo 4

2.3 Pruebas 4

2.4 Capturas de cada prueba 5

**2.4.1: Prueba 1** 5

**2.4.2: Prueba 2** 5

**2.4.3: Prueba 3** 5

**2.4.4: Prueba 4** 5

**2.4.5: Prueba 5** 5

2.5 Implementación 6

Problema 03 7

3.1 Explicación 7

3.2 Algoritmo 8

3.3 Pruebas: 8

3.4 Capturas de cada prueba 9

**3.4.1 Prueba 1** 9

**3.4.2 Prueba 2** 9

**3.4.3 Prueba 3** 9

**3.4.4 Prueba 4** 9

**3.4.5 Prueba 5** 10

**3.4.6 Prueba 6** 10

3.5 Implementación 11

Problema 04 12

4.1 Explicación 12

4.2 Algoritmo 12

4.3 Pruebas 12

4.4 Capturas de cada prueba 13

**4.4.1 Prueba 1** 13

**4.4.2 Prueba 2** 13

**4.4.3 Prueba 3** 13

**4.4.4 Prueba 4** 14

4.5 Implementación 14

Problema 05 15

5.1 Explicación 15

5.2 Algoritmo 16

5.3 Pruebas 16

5.4 Capturas de cada prueba 17

**5.4.1 Prueba 1** 17

**5.4.2 Prueba 2** 17

**5.4.3 Prueba 3** 17

5.5 Implementación 18

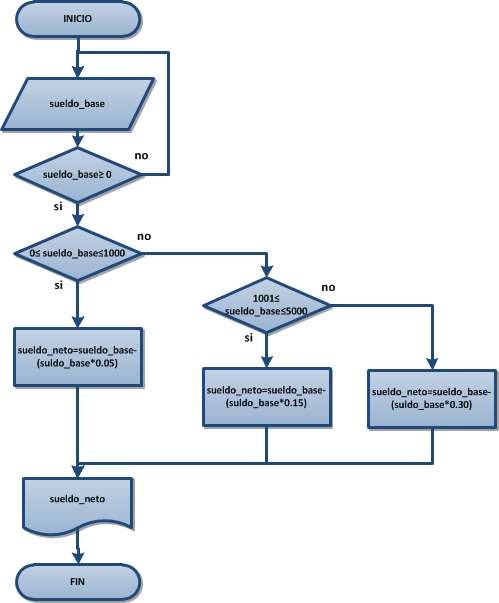
# **Problema 02**

**Descripción**: Se necesita realizar un programa que calcule el sueldo neto de un trabajador, el programa recibirá el sueldo base del empleado y le descontará el impuesto sobre la renta según la siguiente tabla

|  |  |
| --- | --- |
| Rangos del sueldo base | Descuento del impuesto |
| $0.00 - $1000.00 | 5% del sueldo base |
| $1001.00 - $5000.00 | 15% del sueldo base |
| >$5000 | 30% del sueldo base |

**2.1 Explicación:**

Primeramente, en este programa de inicio, el usuario debe de ingresar el valor de su sueldo base, pero el programa solo realizará el cálculo si y solo si el número ingresado es mayor o igual a cero, si el número es mayor a cero entonces el programa realizará el cálculo correspondiente, claro, dependiendo del rango en el que esté, cosa que se ha mostrado en la tabla anterior. Tenemos una sentencia de selección, en donde se evalúa si el valor ingresado esta entre 1000 y 0, personalmente creo que el especificar que está desde 0 es innecesario ya que el programa no llegará hasta este punto si el número es menor a 0, si esta sentencia se cumple, entonces, se realizará una operación aritmética muy sencilla, que será, el sueldo base menos, el producto del sueldo base por el porcentaje que se restará, si es 5% se multiplica por 0.05 si es 15% se multiplicará por 0.15, si es 30% la multiplicación es por 0.30. La segunda sentencia, evalúa si el rango del número ingresado esta entre 1001 y 5000 si se cumple entonces el porcentaje a restar es 15%, finalmente si ninguna de las sentencias anteriores se cumple, el porcentaje restado será del 30%. Una vez realizado lo anterior, se le muestra al usuario el salario que tendrá al restarle el ISR. Cabe resaltar que este programa, al menos en mi computadora, ingrese el valor que sea, solo hace la segunda evaluación (En el rango de 1001 a 5000), creo que, en cuanto a la lógica de programación, está de manera correcta, pero claro, podría estar equivocado.

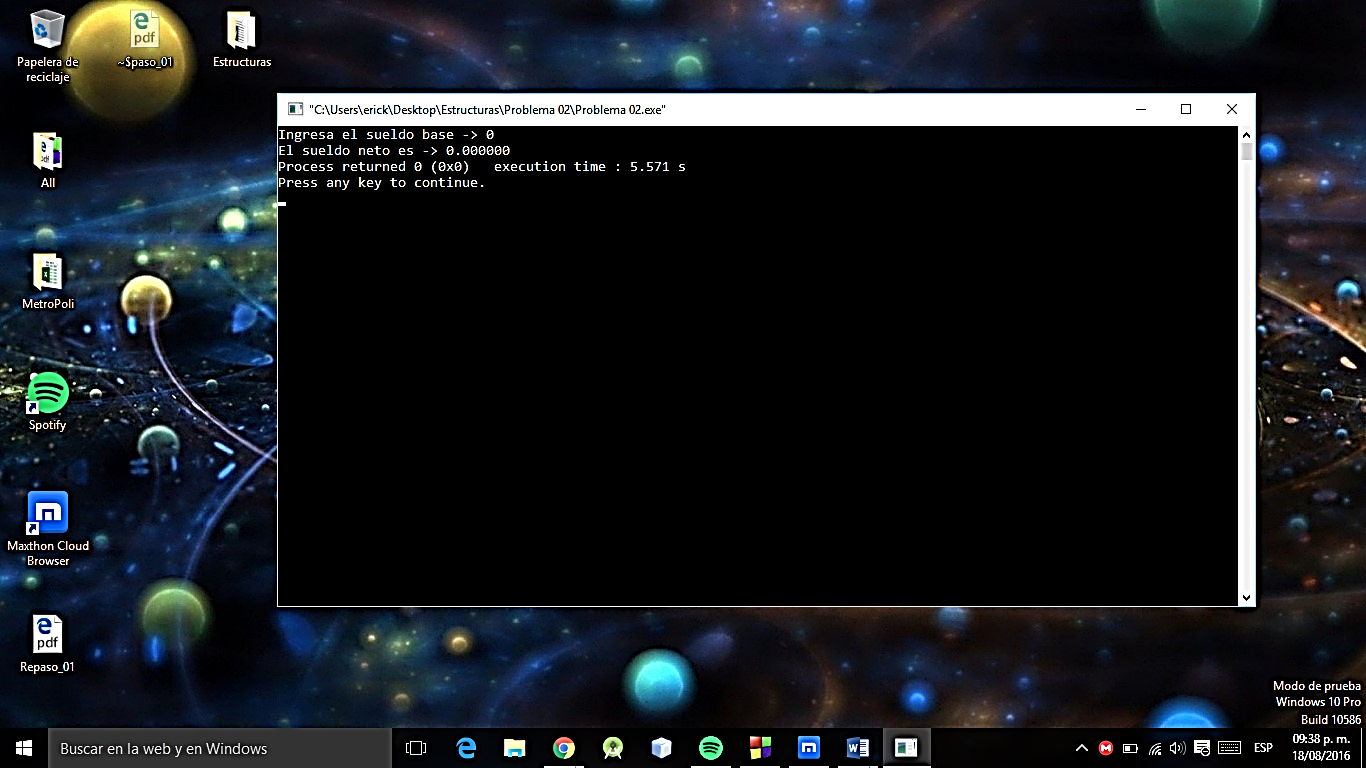
******2.2 Algoritmo:**

**2.3 Pruebas:**

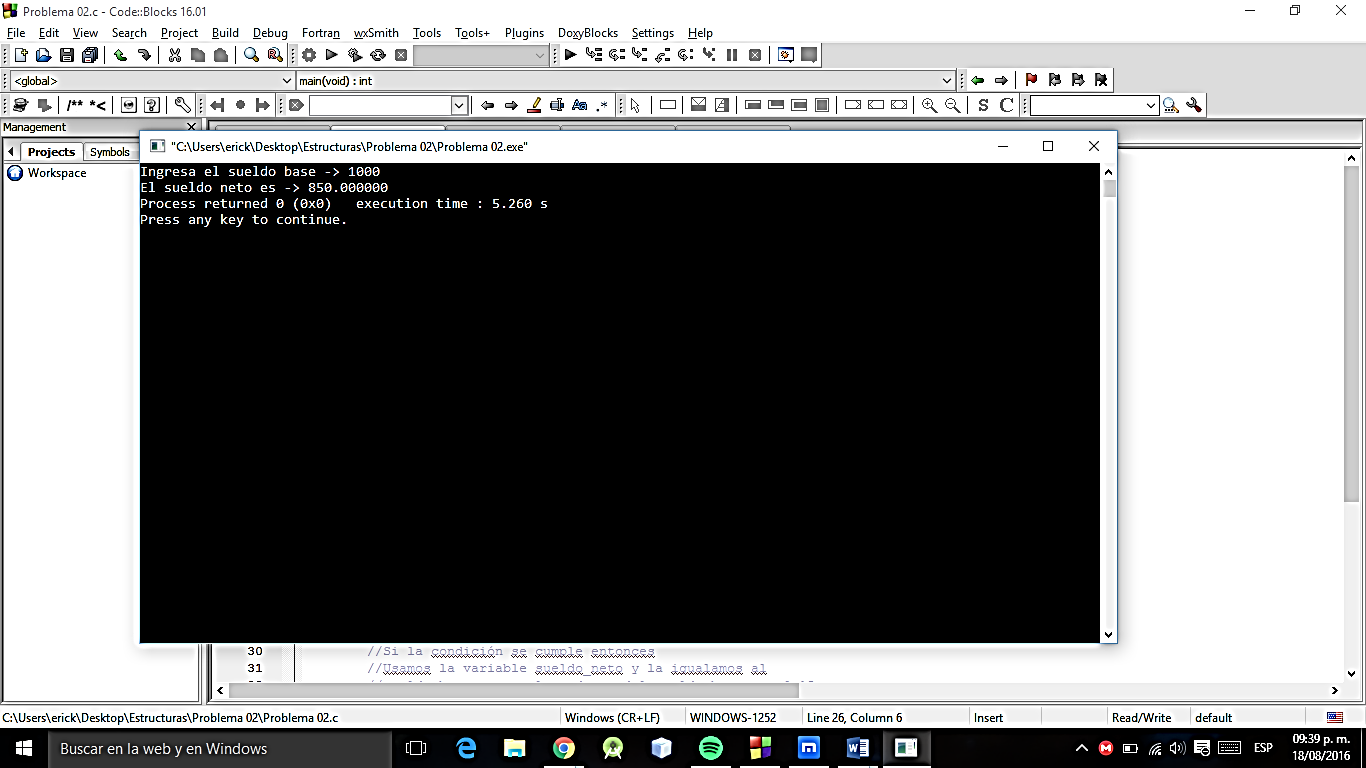
|  |  |  |  |
| --- | --- | --- | --- |
| # Prueba | Entrada | Salida | ¿Es el correcto? |
| 1 | **0** | **0** | **Si** |
| 2 | **1000** | **850** | **No** |
| 3 | **500** | **425** | **No** |
| 4 | **2500** | **2125** | **Si** |
| 5 | **10000** | **8500** | **No** |

**2.4 Capturas de cada prueba:**

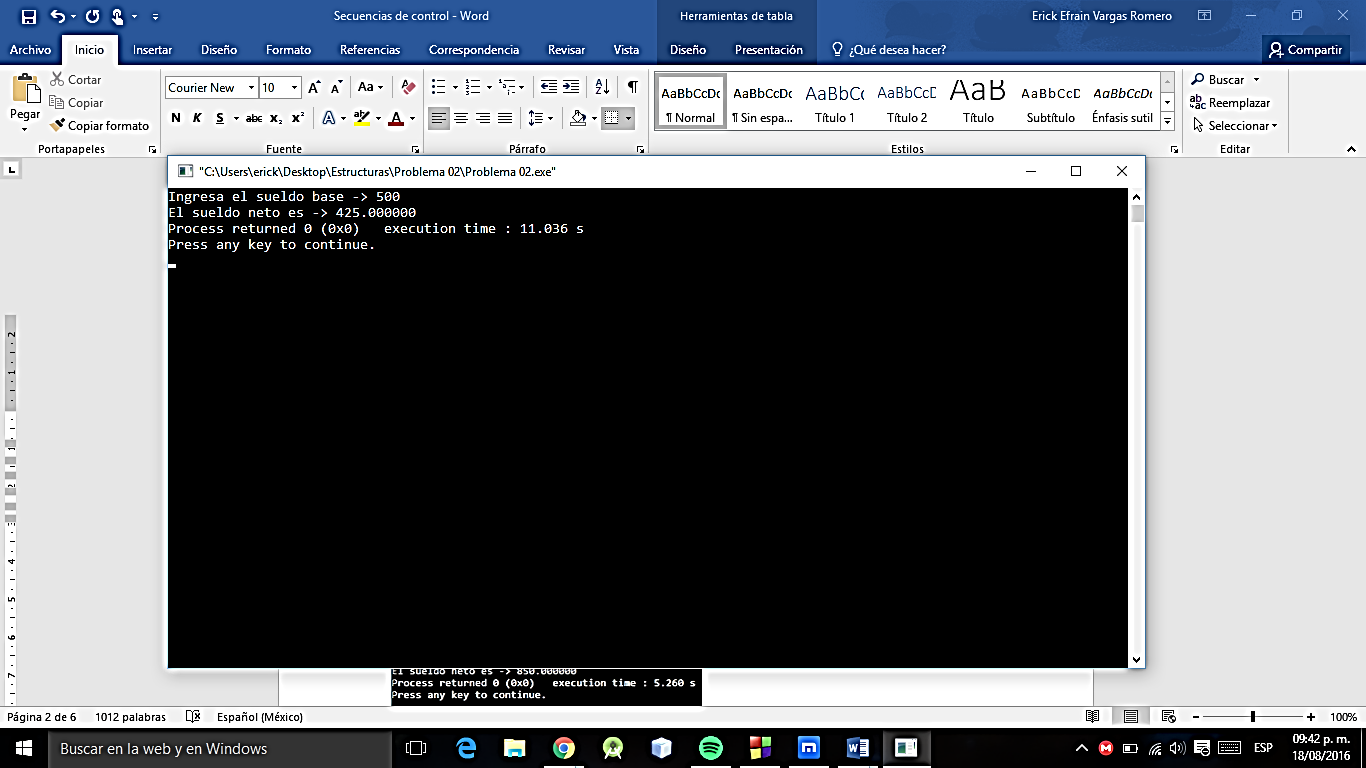
### **2.4.1: Prueba 1**



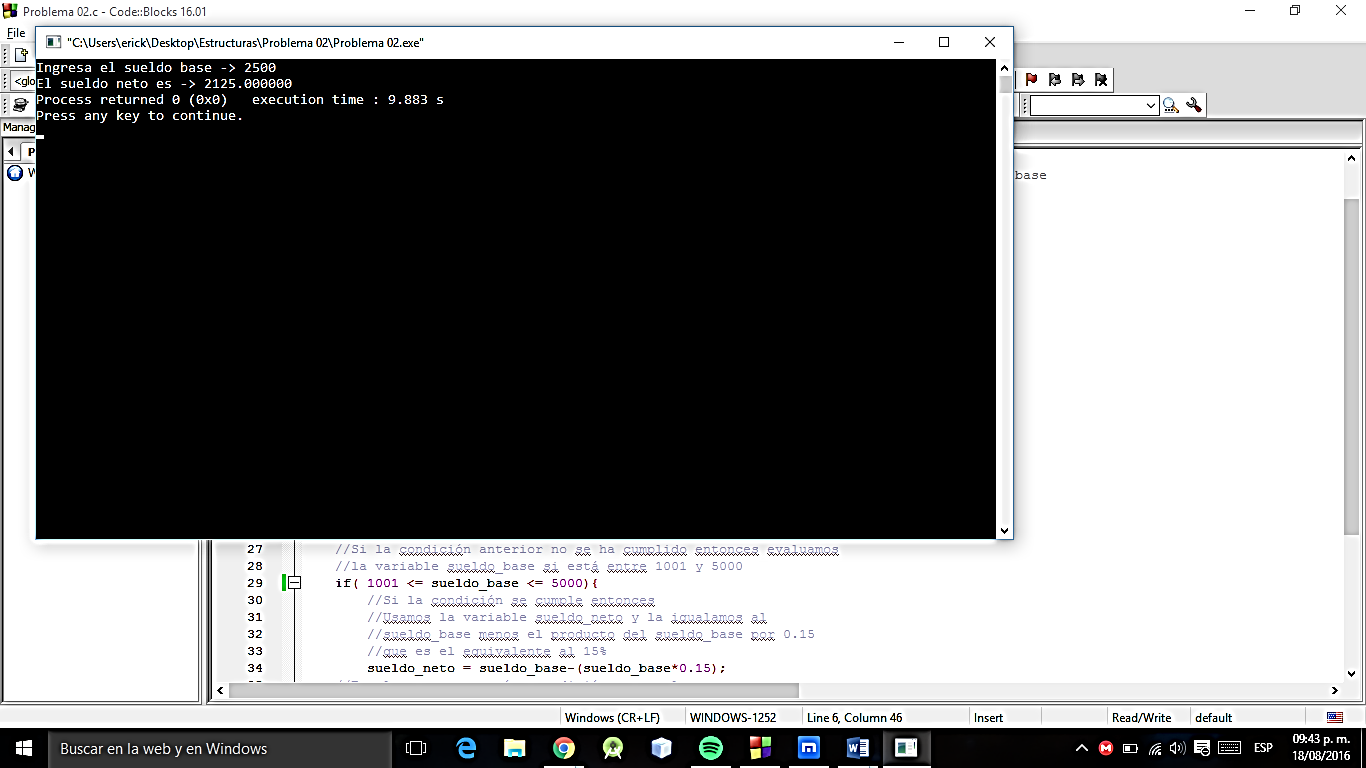
### **2.4.2: Prueba 2**



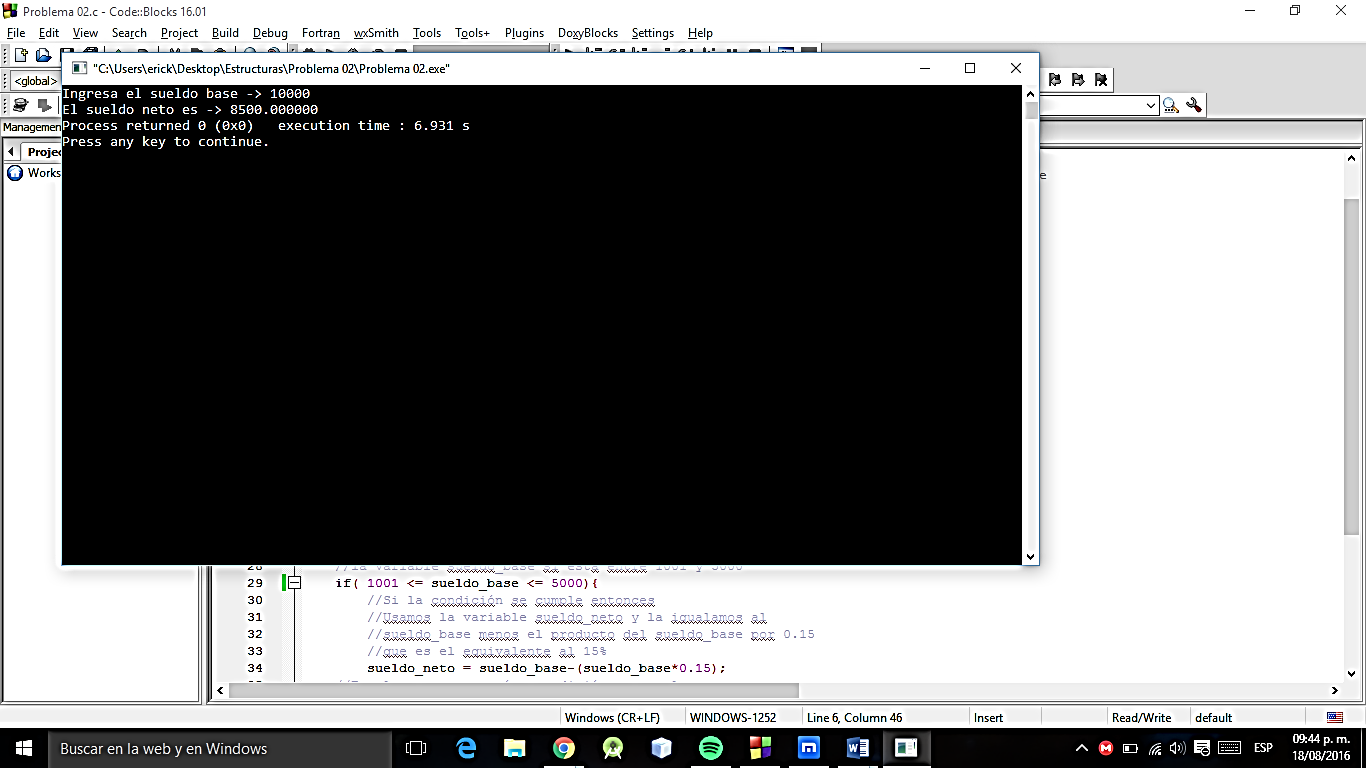
### **2.4.3: Prueba 3**



### **2.4.4: Prueba 4**



### **2.4.5: Prueba 5**



# **2.5 Implementación:**

#include <stdio.h> //Librerias, se incluyen las funciones estandar de entrada y salida

//Programa principal

int main**(**void**){**

/\*Declaración de dos variables una donde se almacena el sueldo neto y otra el sueldo base

el suedo neto es el sueldo menos el ISR\*/

float sueldo\_base **=** 0**,** sueldo\_neto **=** 0**;**

//Inicio del ciclo de repetición

**do{**

//Enviamos un mensaje al usuario

printf**(**"Ingresa el sueldo base -> "**);**

/\*Lectura del dato que ingresará el usuario, y es almacenado

en la variable sueldo base\*/

scanf**(**"%f"**,** **&**sueldo\_base**);**

//Fin del ciclo de repetición mientras el sueldo base no sea mayor-igual a cero

**}while(!(**sueldo\_base**>=**0**));**

//Evaluamos si el valor del sueldo\_base esta entre 0 y 1000

**if(** 0 **<=** sueldo\_base **<=** 1000**){**

// Si se cumple la condición entonces

//Usamos la variable sueldo\_neto y la igualamos al

//sueldo\_base menos el producto del sueldo\_base por 0.05

//que es el equivalente al 5%

sueldo\_neto **=** sueldo\_base**-(**sueldo\_base**\***0.05**);**

**}**

//Si la condición anterior no se ha cumplido entonces evaluamos

//la variable sueldo\_base si está entre 1001 y 5000

**if(** 1001 **<=** sueldo\_base **<=** 5000**){**

//Si la condición se cumple entonces

//Usamos la variable sueldo\_neto y la igualamos al

//sueldo\_base menos el producto del sueldo\_base por 0.15

//que es el equivalente al 15%

sueldo\_neto **=** sueldo\_base**-(**sueldo\_base**\***0.15**);**

//En el caso que nungúna condición se cumpla

**}** **else** **{**

//Usamos la variable sueldo\_neto y la igualamos al

//sueldo\_base menos el producto del sueldo\_base por 0.30

//que es el equivalente al 30%

sueldo\_neto **=** sueldo\_base**-(**sueldo\_base**\***0.30**);**

**}**

//Mandamos un mensaje a la pantalla con el valor del sueldo neto

printf**(**"El sueldo neto es -> %f"**,** sueldo\_neto**);**

**return** 0**;**

**}**

# **Problema 03**

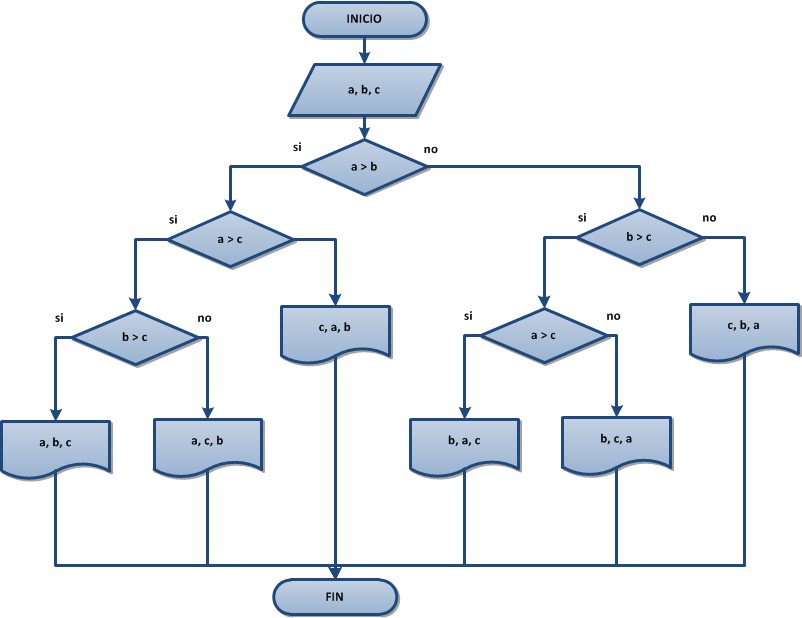
**Descripción**: dados tres números enteros de entrada a, b y c, mostrarlos a la salida ordenados de forma descendente.

• E.g. si la entrada es: a=6, b=5, y c=90; La salida será: 90, 6, 5.

• E.g. si la entrada es: a=60, b=50, y c=9; La salida será: 60, 50, 9.

**3.1 Explicación**:

Este programa simplemente nos mostrará de tres números, cual es el mayor, el usuario ingresa el valor de esos tres números, el programa primero evaluará si el primer número es mayor al segundo, si esto se cumple entonces se realiza otra evaluación para saber si el primer número es mayor al tercero si esto se cumple se realiza una última selección en este último caso, y es evaluar si el segundo número es mayor al tercero, si se cumple se muestra el resultado que es, primer número, segundo número y tercer número, en caso contrario se imprime primer número, tercer número y segundo número. Retomando la sentencia de evaluar el primero y el tercero, si el primero no es mayor al tercero entonces se muestra tercero, primero, segundo. Regresemos a la primer evaluación donde se comparaba el primero con el segundo, si el primero no es mayor al segundo entonces se realiza otra evaluación donde comparamos el segundo con el tercero, si se cumple se realiza una última comparación, si el primero es mayor al tercero entonces se muestra, segundo, primero tercero, en caso contrario se muestra segundo, tercero, primero, finalmente si en la comparación de segundo número mayor al tercero no se cumple esta condición el resultado impreso será tercero, segundo primero.

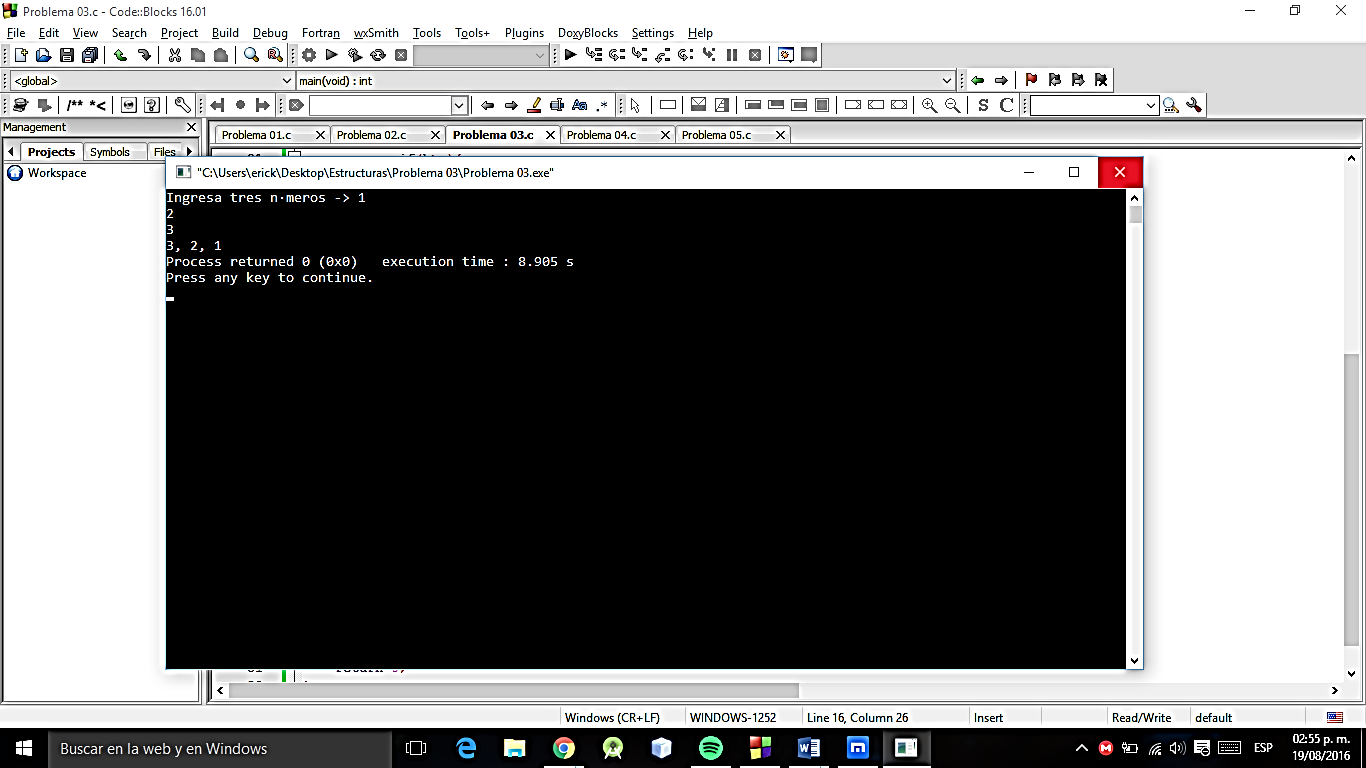
******3.2 Algoritmo:**

## **3.3 Pruebas:**

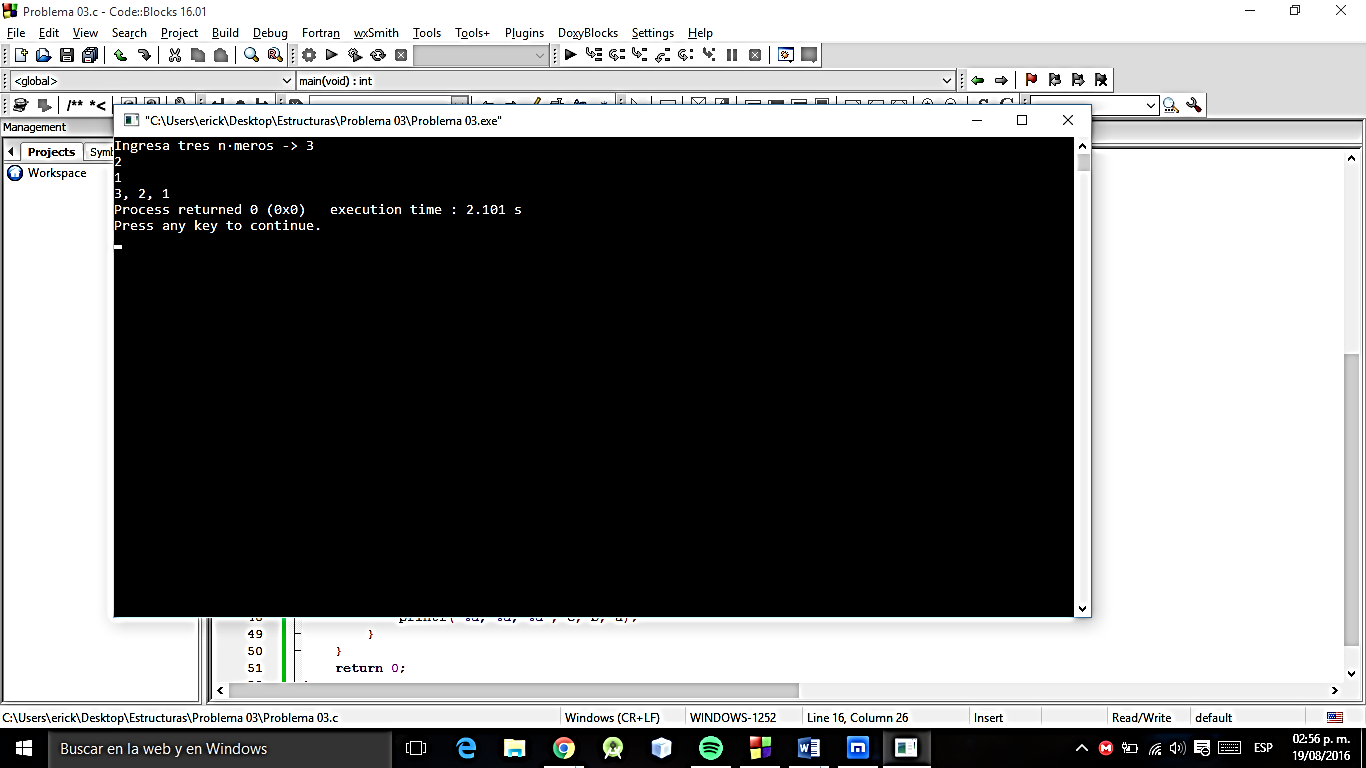
|  |  |  |  |
| --- | --- | --- | --- |
| # Prueba | Entrada | Salida | ¿Es correcto? |
| 1 | 1, 2, 3 | 1, 2, 3 | Si |
| 2 | 3, 2, 1 | 1, 2, 3 | Si |
| 3 | 1, 3, 2 | 1, 2, 3 | Si |
| 4 | 2, 3, 1 | 1, 2, 3 | Si |
| 5 | 3, 1, 2 | 1, 2, 3 | Si |
| 6 | 2, 1, 3 | 1, 2, 3 | Si |

## **3.4 Capturas de cada prueba**

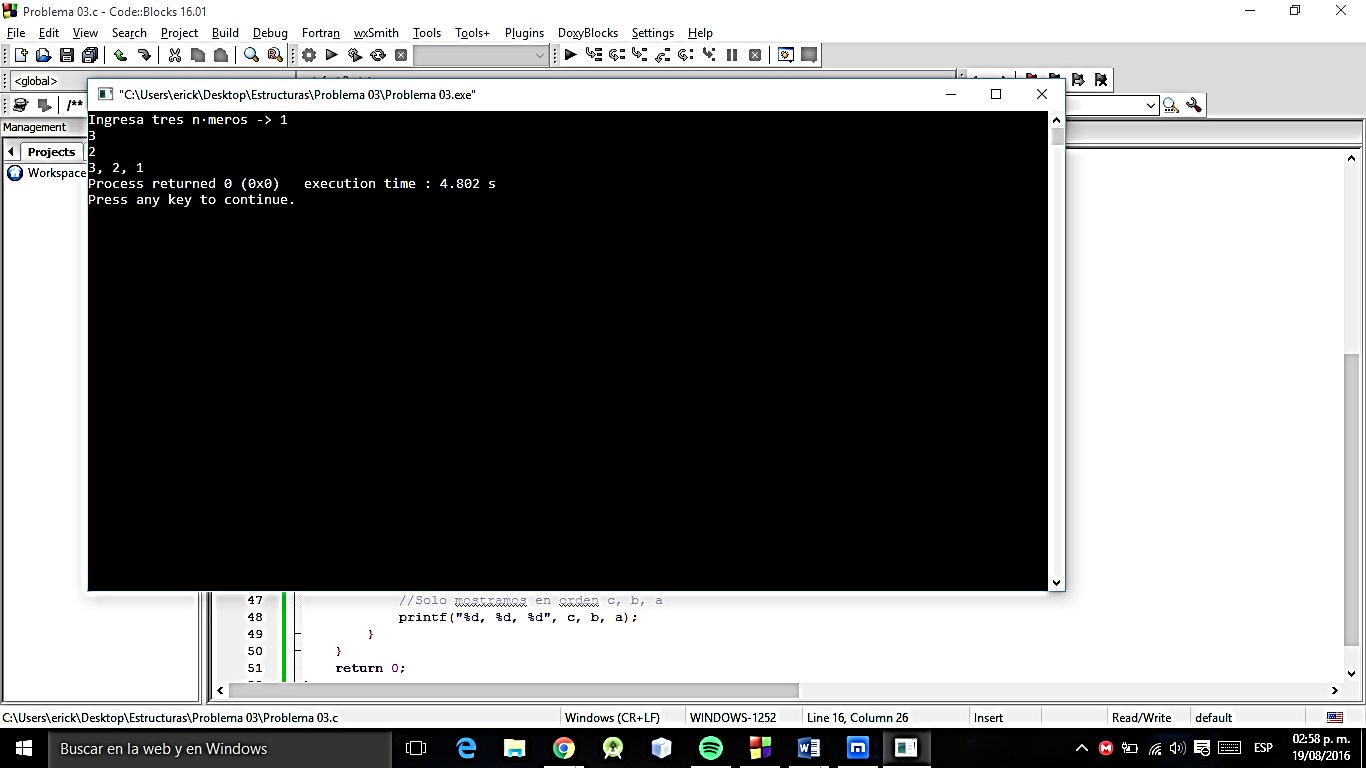
### **3.4.1 Prueba 1:**



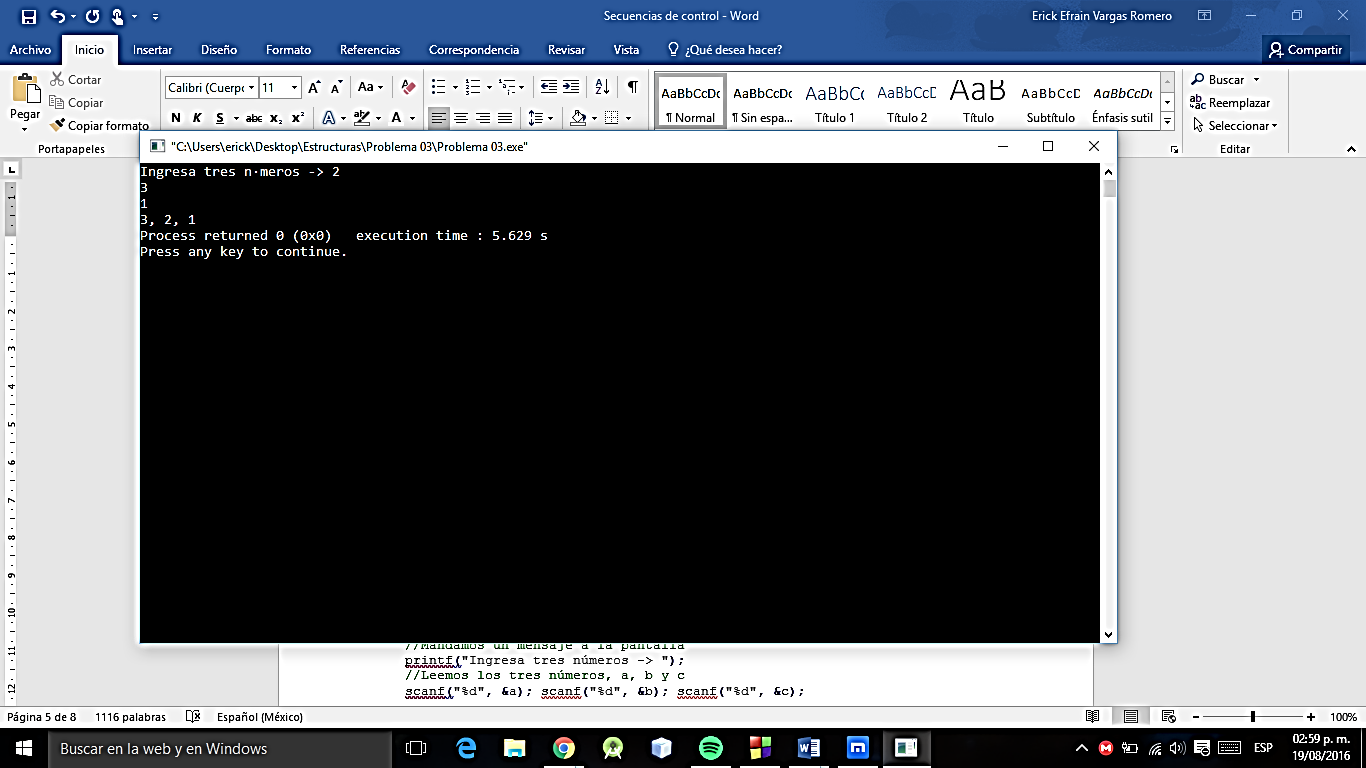
### **3.4.2 Prueba 2:**



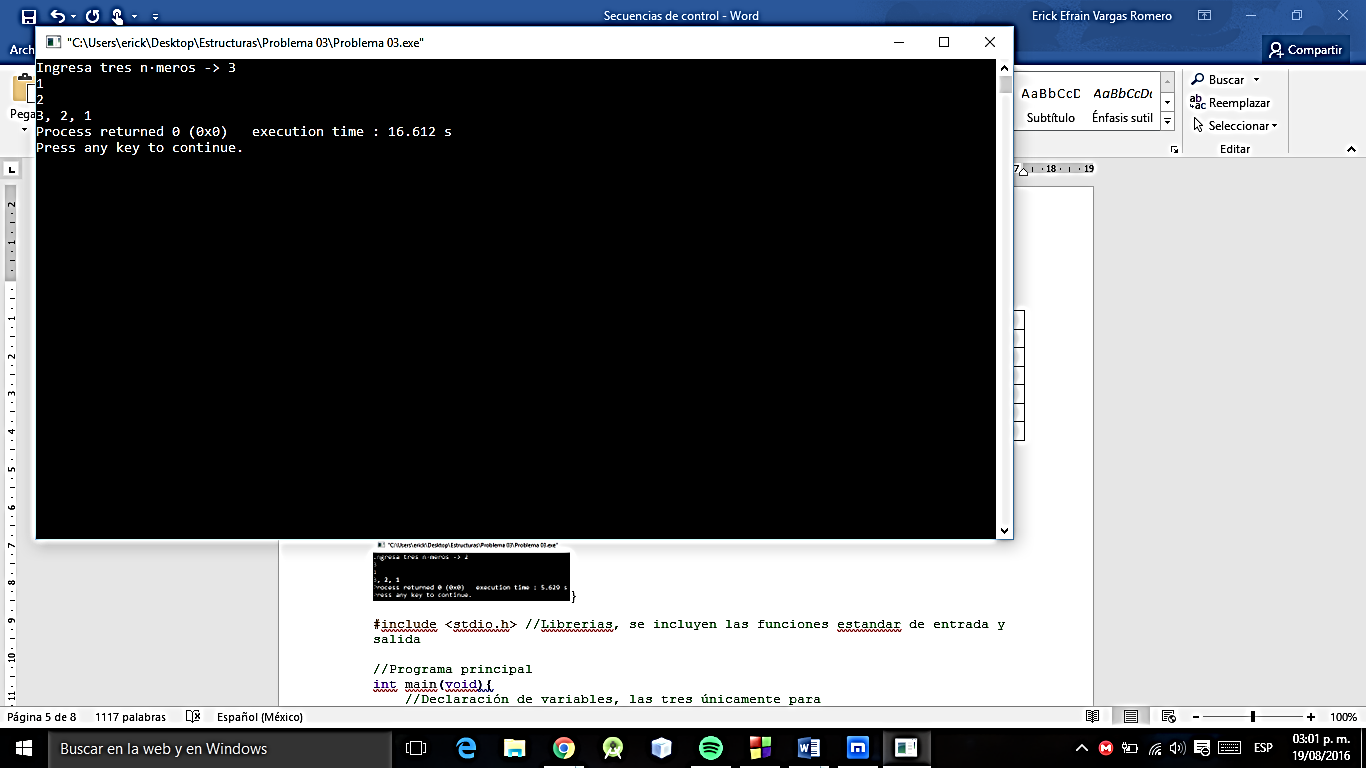
### **3.4.3 Prueba 3:**



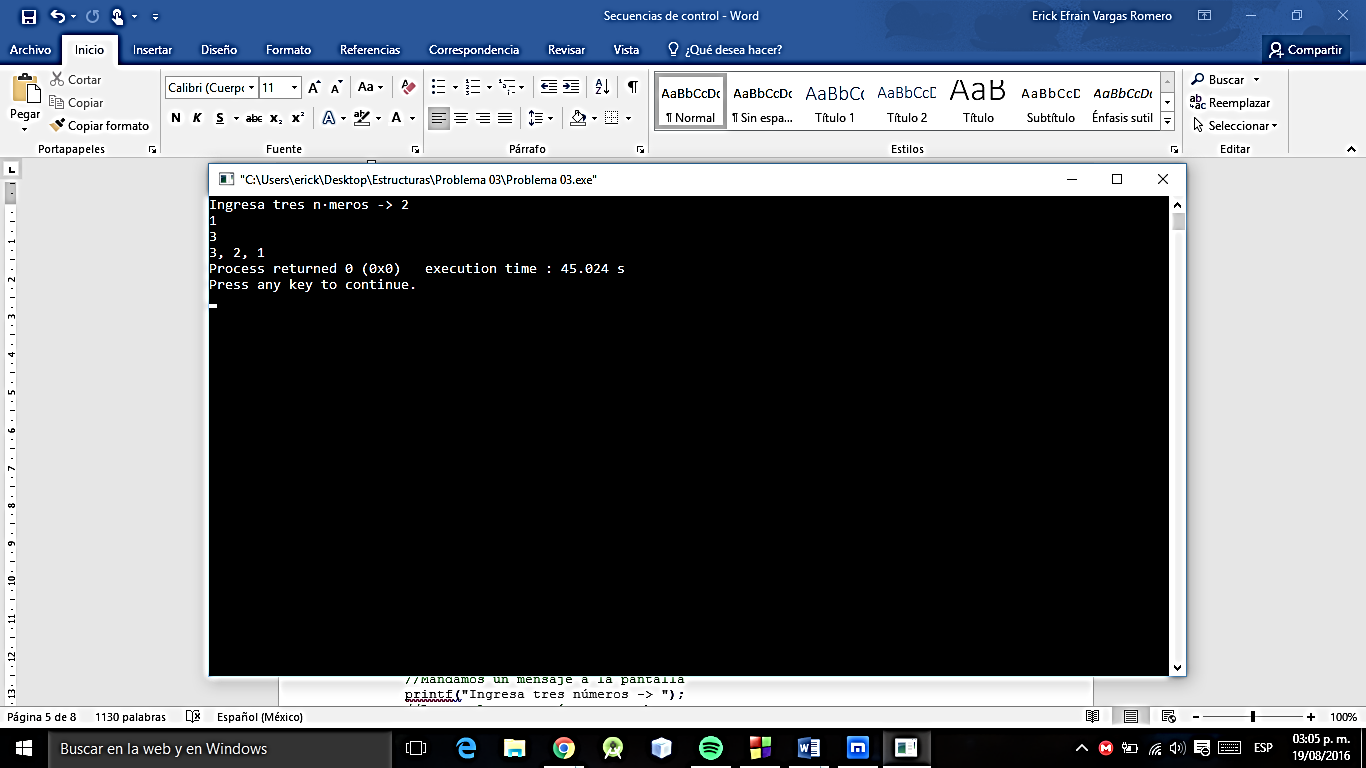
### **3.4.4 Prueba 4:**



### **3.4.5 Prueba 5:**



### **3.4.6 Prueba 6:**



## **3.5 Implementación**

#include <stdio.h> //Librerias, se incluyen las funciones estandar de entrada y salida

//Programa principal

int main**(**void**){**

//Declaración de variables, las tres únicamente para

//almacenar el valor de tres números

int a **=** 0**,** b **=** 0**,** c **=** 0**;**

//Mandamos un mensaje a la pantalla

printf**(**"Ingresa tres números -> "**);**

//Leemos los tres números, a, b y c

scanf**(**"%d"**,** **&**a**);** scanf**(**"%d"**,** **&**b**);** scanf**(**"%d"**,** **&**c**);**

//Evaluamos si a es mayor que b

**if(**a**>**b**){**

//Si la condición se cumple volvemos ha

//hacer una evaluación si a es mayor que c

**if(**a**>**c**){**

//Si se cumple la condición, evaluamos nuevamente

//Esta ocasión evaluamos si b es mayor que c

**if(**b**>**c**){**

//Si la condición se cumple, imprimimos los

//tres números primero a, luego b y c

printf**(**"%d, %d, %d"**,** a**,** b**,** c**);**

**}** **else** **{** //Si b es menor que C entonces

//Solo mostramos el orden a, c, y b

printf**(**"%d, %d, %d"**,** a**,** c**,** b**);**

**}**

**}** **else** **{** //Si a resulta ser menor que c

//Mostramos el orden c, a, b

printf**(**"%d, %d, %d"**,** c**,** a**,** b**);**

**}**

**}** **else** **{** //Si a es menor que b, entonces

//evaluamos nuevamente si b es mayor que c

**if(**b**>**c**){**

//Si la condición se cumple, entonces evaluamos

//si a es mayor que C

**if(**a**>**c**){**

//Si la condición se cumple, mostramos

//en el orden, b primero, segundo a y final c

printf**(**"%d, %d, %d"**,** b**,** a**,** c**);**

**}** **else** **{** //Si a resulta ser menor que c entonces

//mostramos el orden, b, c, a

printf**(**"%d, %d, %d"**,** b**,** c**,** a**);**

**}**

**}** **else** **{** //Si b es menor que c

//Solo mostramos en orden c, b, a

printf**(**"%d, %d, %d"**,** c**,** b**,** a**);**

**}**

**}**

**return** 0**;**

**}**

# **Problema 04:**

**Descripción**: Realizar un programa que calcule el cubo de una serie de números positivos introducidos uno a uno, si se introduce un número negativo se deberá terminar el programa. El primer número a introducir nunca será negativo.

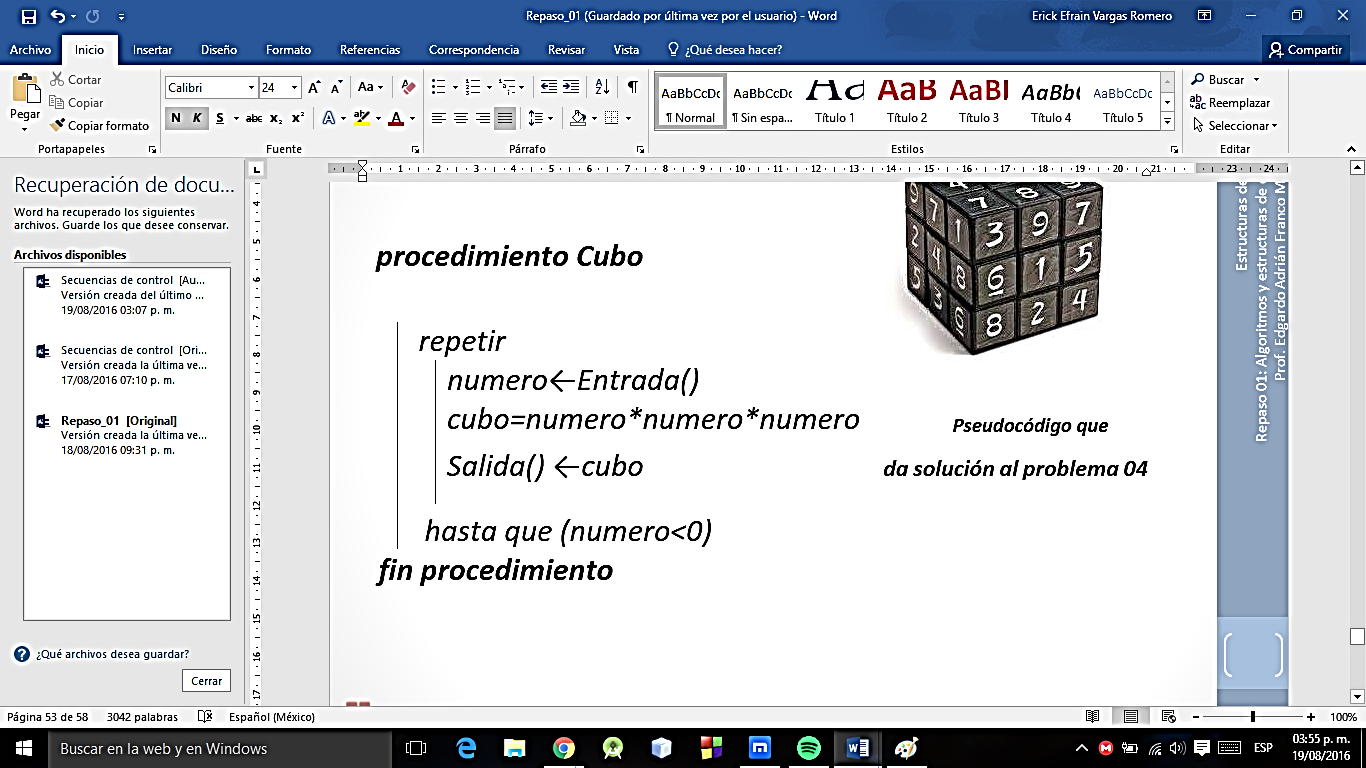
**4.1 Explicación:**

Este programa es realmente sencillo de explicar, el usuario ingresará un número, pero este número solo debe de ser positivo, si el número es negativo, el programa finalizará. Si el número es mayor a cero, elevaremos al cubo este número, recordemos que el cubo de un número es el multiplicar por sí mismo ese número tres veces, por ejemplo:

23 => 2x2 = 4; 4x2 = 8

Una vez realizada la operación anterior, mostramos el resultado al usuario, y el usuario tiene la posibilidad de ingresar otro número, pero como se mencionó anteriormente, si el número ingresado es negativo, el programa finalizará.

**4.2 Algoritmo:**



## 

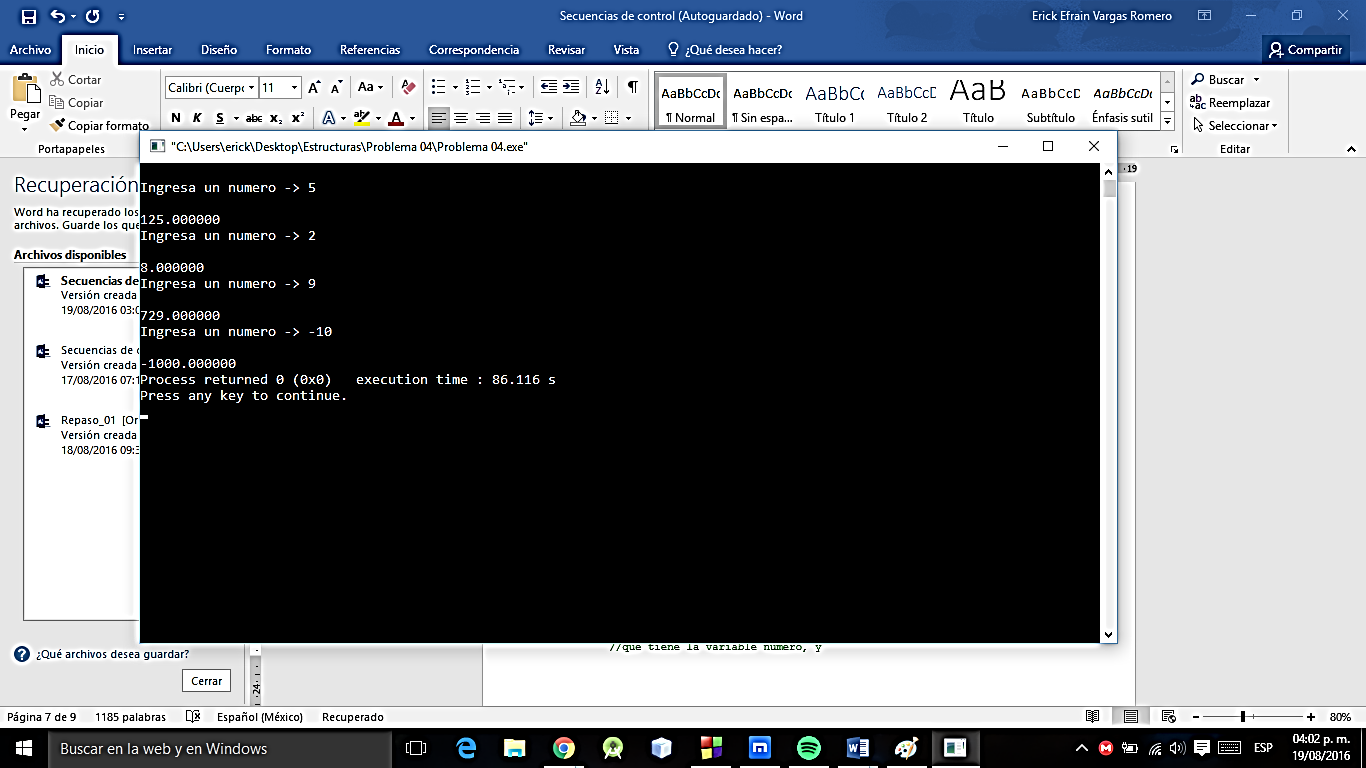
## **4.3 Pruebas:**

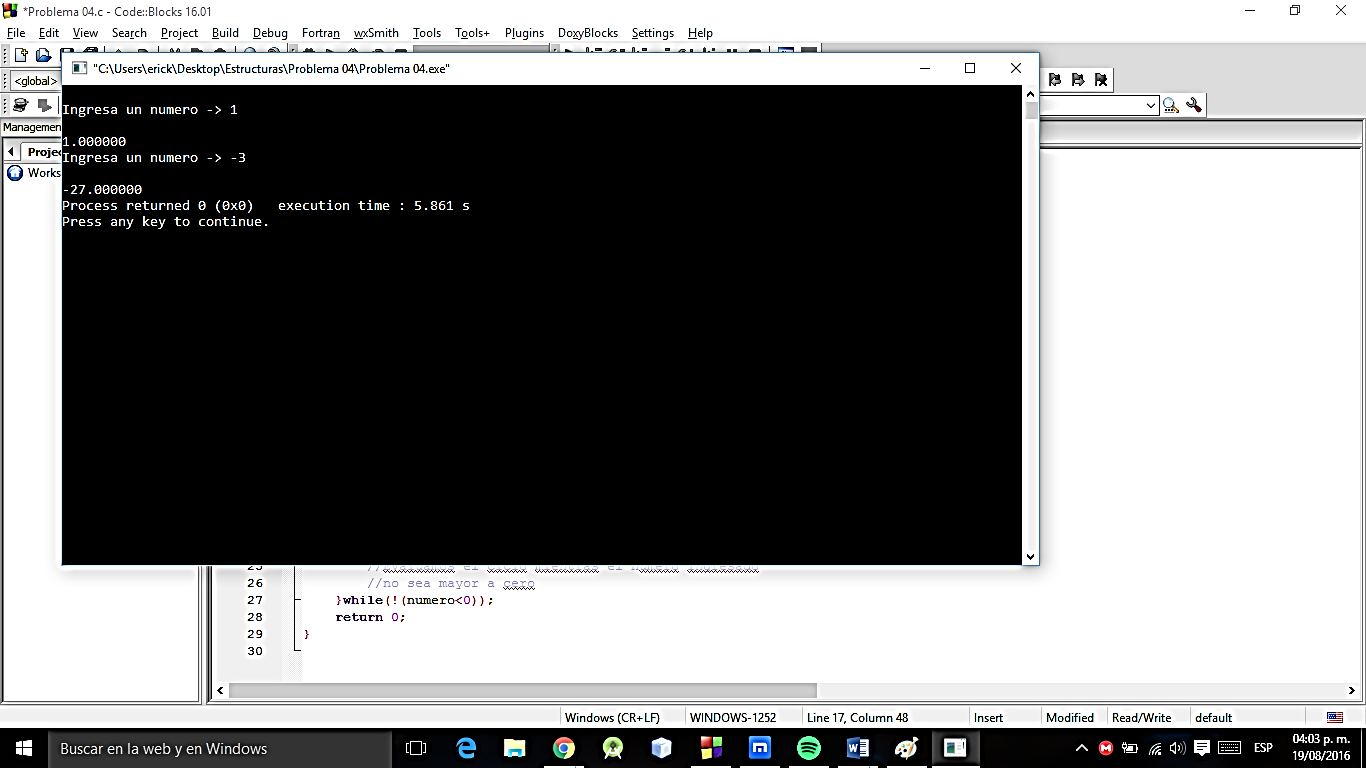
|  |  |  |  |
| --- | --- | --- | --- |
| # Prueba | Entrada | Salida | ¿Es correcto? |
| 1 | 5, 2, 9, -10 | 125, 8, 729, -1000 | Si |
| 2 | 1, -3 | 1, -27 | Si |
| 3 | -3 | -27 | Si |
| 4 | 1, 2, 3, 4, 5, 6, -1000 | 1, 8, 27, 64, 125, 216, -1000000 | Si |

## 

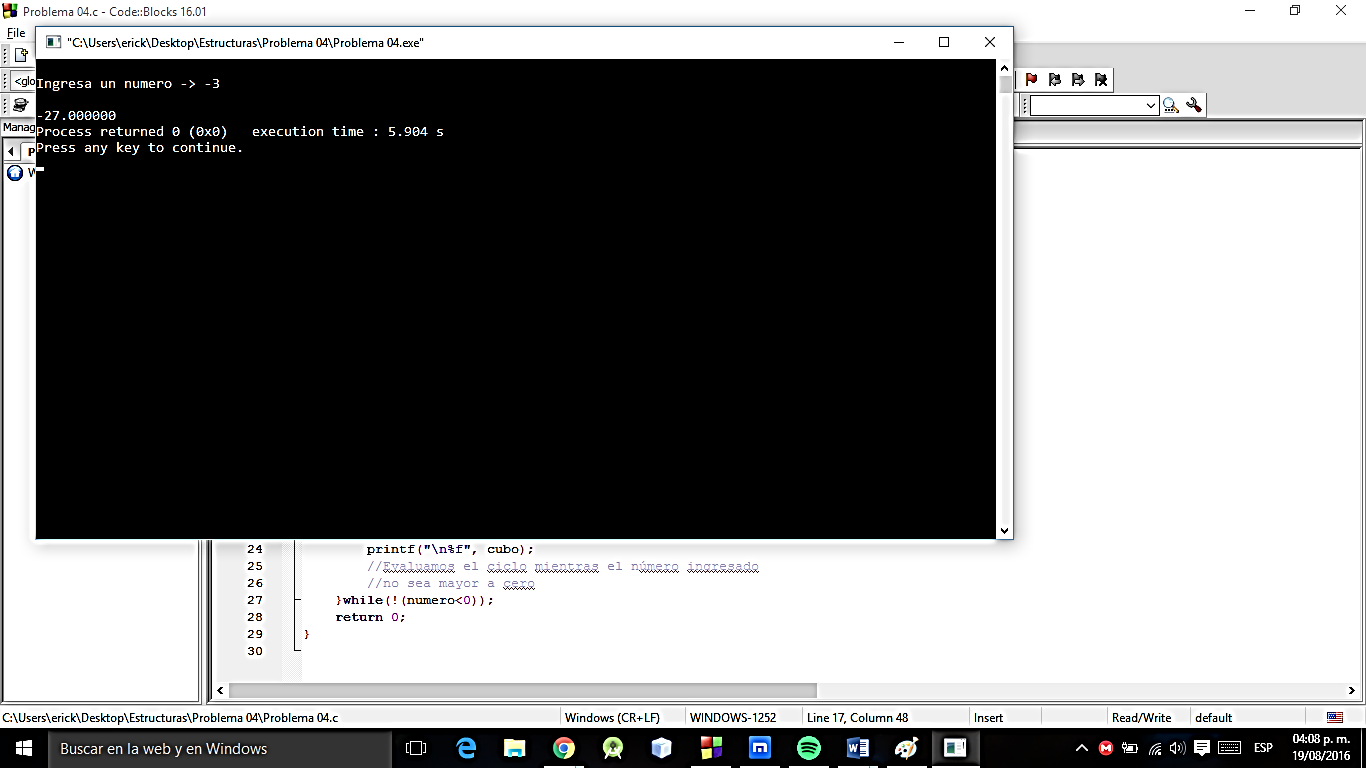
## **4.4 Capturas de cada prueba:**

### **4.4.1 Prueba 1:**

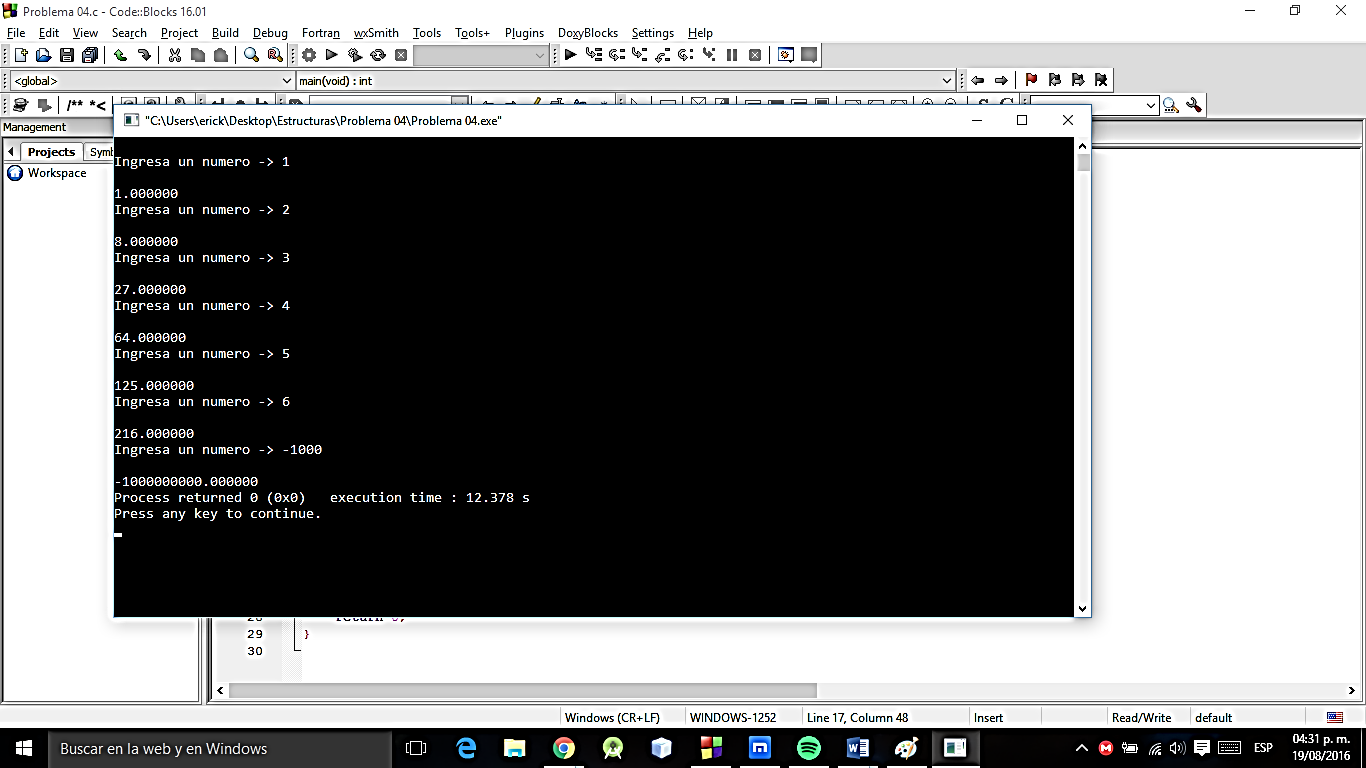


******4.4.2 Prueba 2:**

**4.4.3 Prueba 3:**



**4.4.4 Prueba 4:**



**4.5 Implementación:**

#include <stdio.h> //Librerias, se incluyen las funciones estandar de entrada y salida

//Programa principal

int main**(**void**){**

//Declaración de variables, la primera

//para almacenar el número que ingresará el usuario

//la segunda almacena el valor de la primer variable

//pero multiplicada por si misma 3 veces

float numero **=** 0**,** cubo**;**

//Inicio del ciclo de repetición

**do{**

//Mostramos un mensaje

printf**(**"\nIngresa un numero -> "**);**

//Hacemos la lectura y almacenamiento del

//numero del usuario en la variable numero

scanf**(**"%f"**,** **&**numero**);**

//Almacenamos en la variable cubo el valor

//que tiene la variable número, y

//la multiplicamos por si misma tres veces

cubo **=** numero**\***numero**\***numero**;**

//Mostramos el valor de la variable cubo

printf**(**"\n%f"**,** cubo**);**

//Evaluamos el ciclo mientras el número ingresado

//no sea mayor a cero

**}while(!(**numero**<**0**));**

**return** 0**;**

**}**

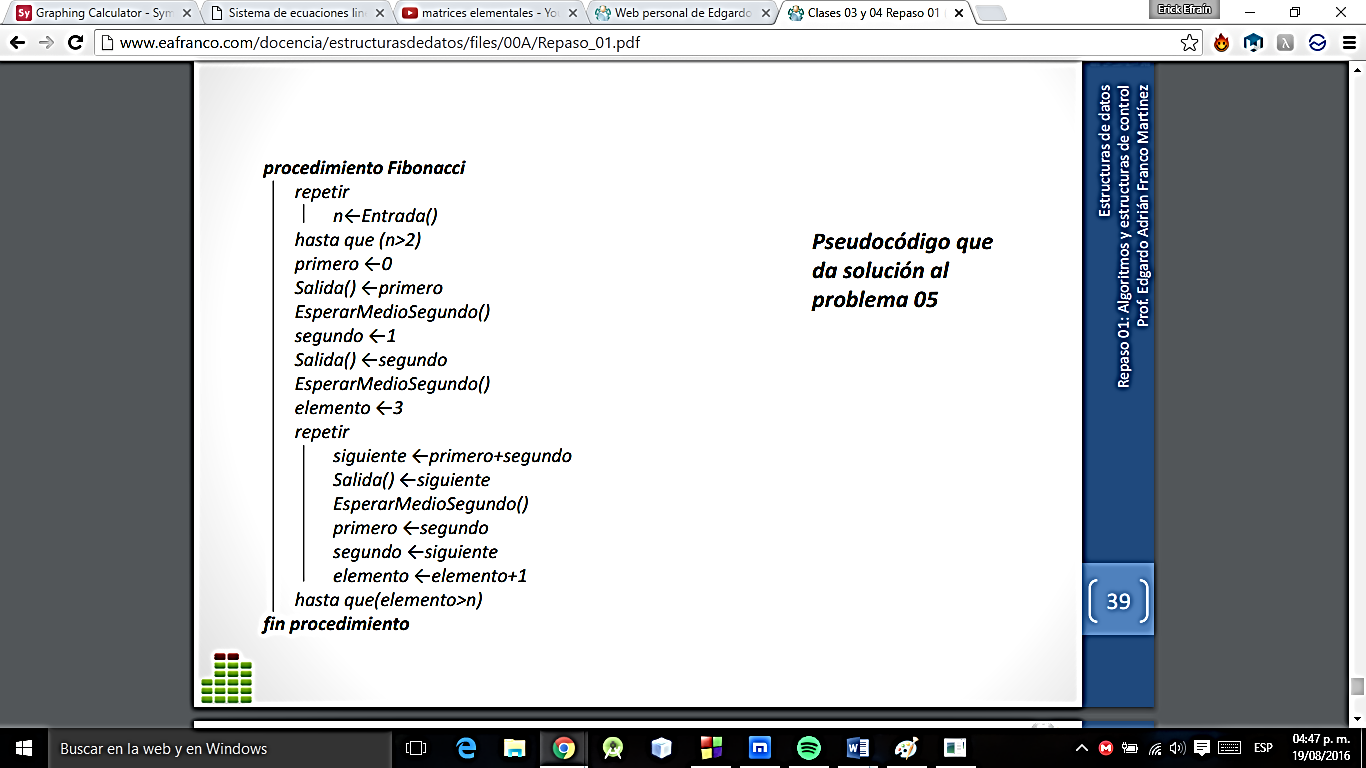
# **Problema 05**

**Descripción:** Realizar un programa que muestre la serie de Fibonacci hasta el término n, el cual es recibido al inicio y debe ser mayor a 2. Al mostrar la serie se deberá mostrar termino a término a la salida con un retraso de 0.5 segundos.

# **5.1 Explicación:**

Un programa sencillo de explicar, en el cual el usuario puede ingresar cualquier número, pero si este número no es mayor a dos no se mostrará la sucesión, si el número es mayor a tres entonces se almacenarán los primeros valores de la sucesión que son 0, y 1 entre cada valor que se mostrará el programa espera medio segundo. Una vez mostrados los dos valores anteriores entramos a un ciclo de repetición, al iniciar el siclo tenemos un variable donde almacenaremos el valor de la suma de los dos primeros números ya antes mostrados, y lo mostrará al usuario, posteriormente se esperará medio segundo, y ahora realizamos una igualación, que será el primer número, ahora tendrá el valor del segundo, el segundo tendrá el valor de la sumatoria que se realizó previamente, y finalmente realizaremos un conteo, de las veces que se ha hecho todo lo anterior, este ciclo terminará cuando el contador que acabamos de mencionar, es mayor al número ingresado inicialmente por el usuario.

**5.2 Algoritmo:**

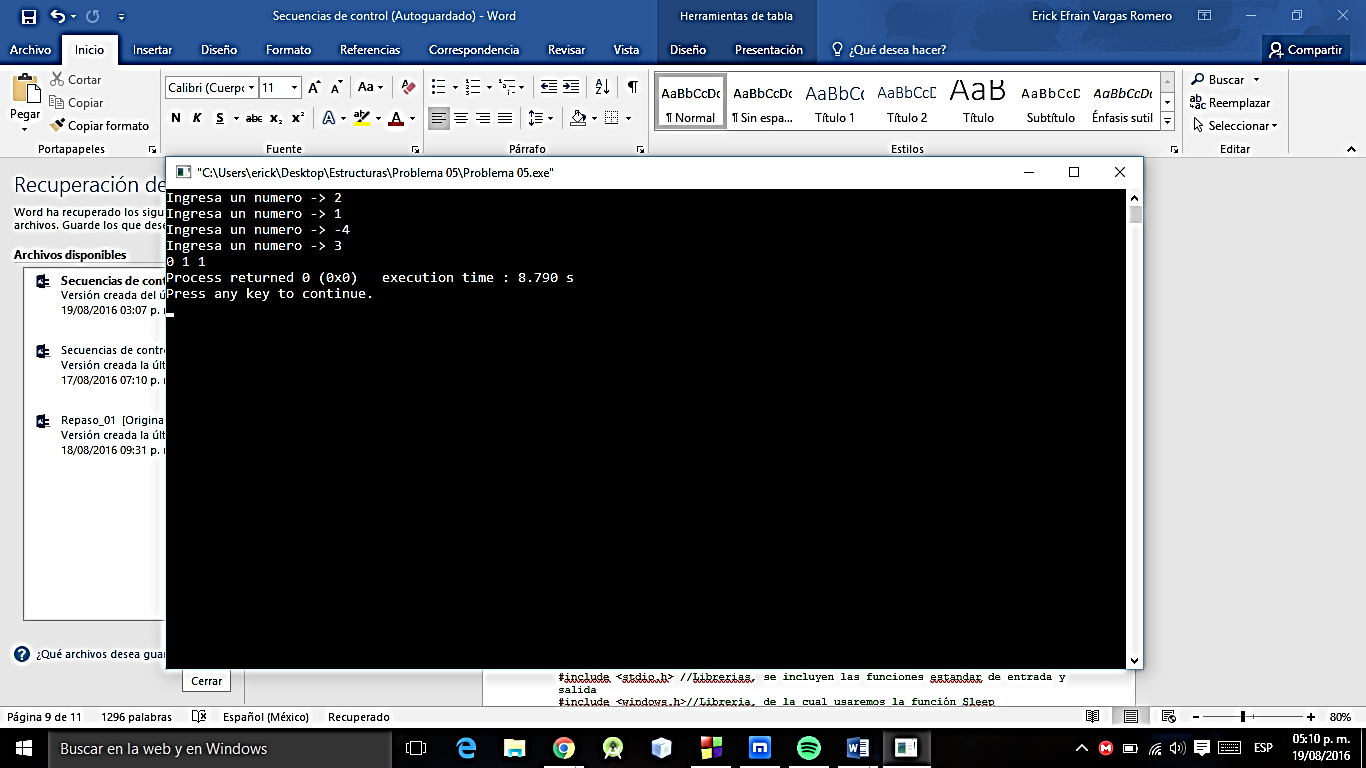


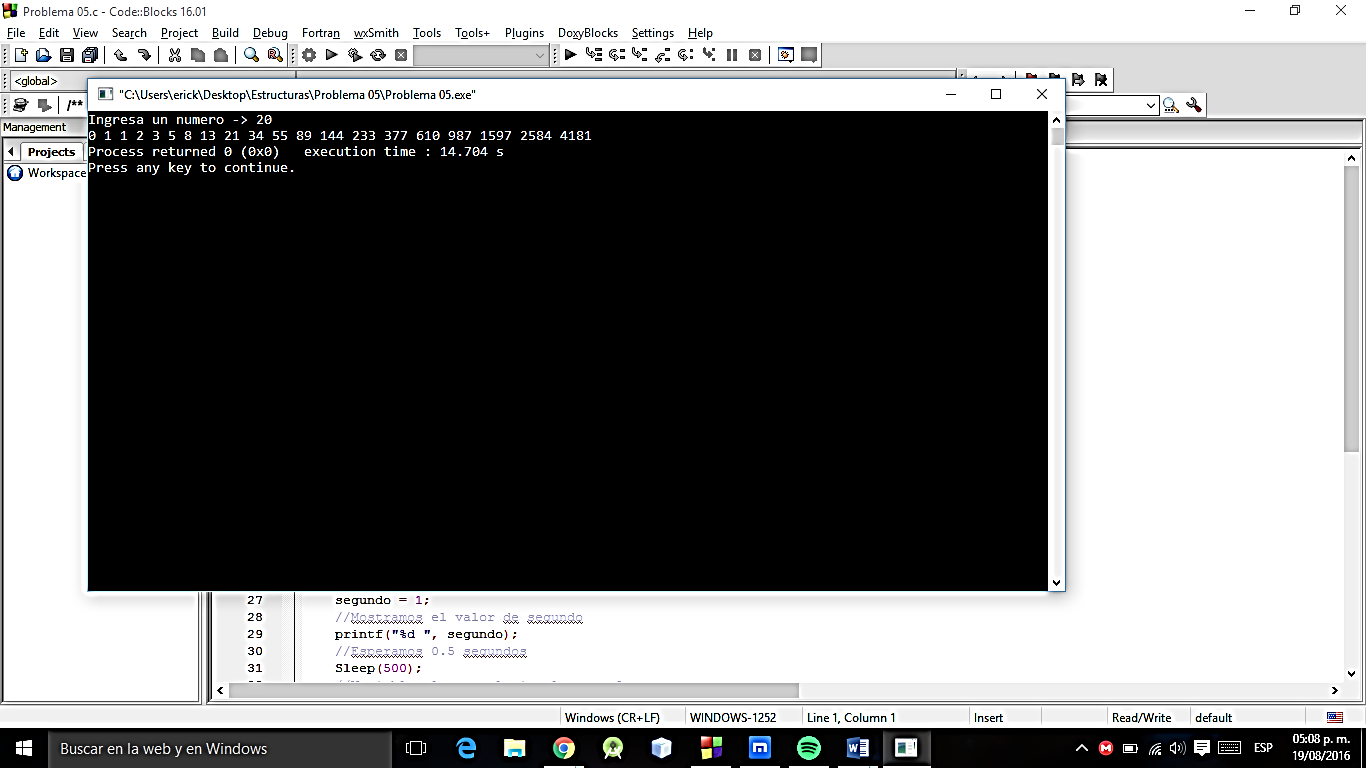
**5.3 Pruebas:**

|  |  |  |  |
| --- | --- | --- | --- |
| Prueba número: | Entrada | Salida | ¿Es correcto? |
| 1 | 2, 1, -4, 3 | 1 | Si |
| 2 | 20 | 4181 | Si |
| 3 | 8 | 13 | Si |

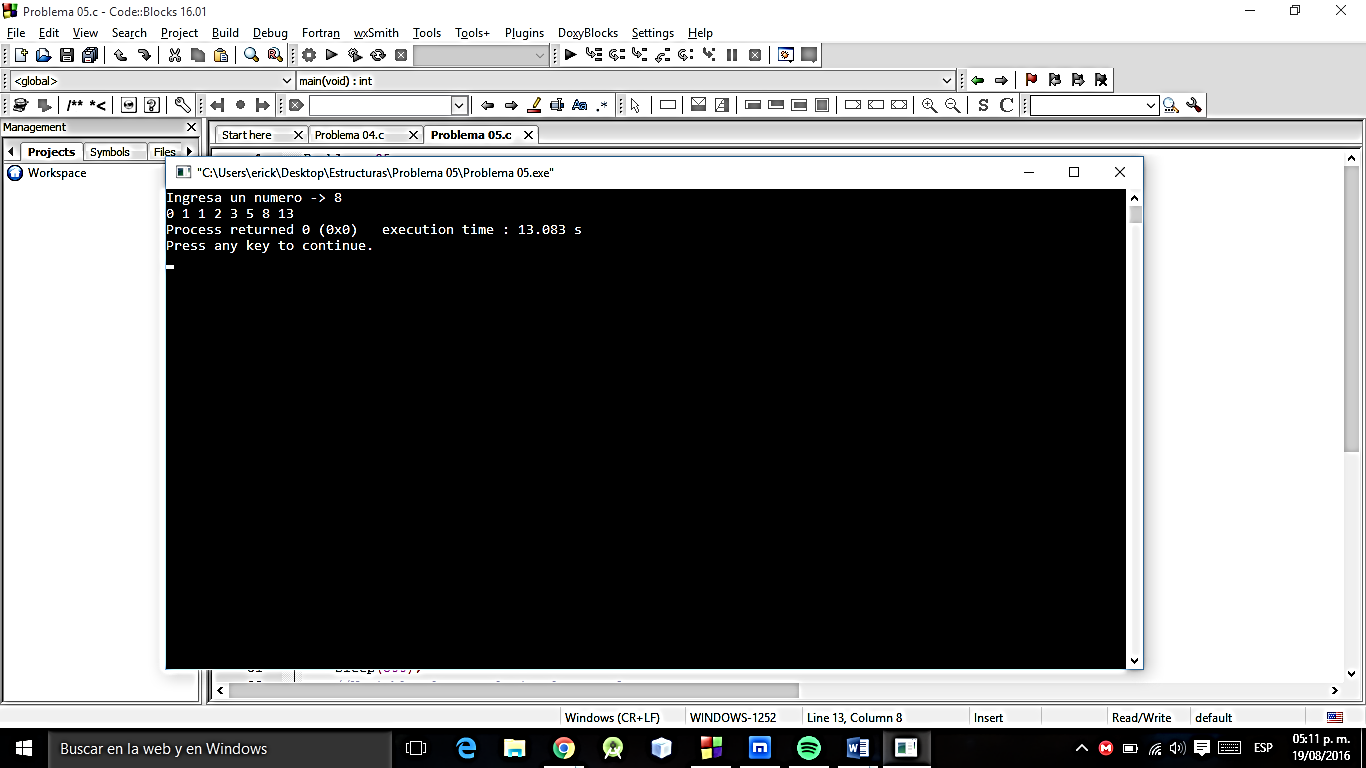
**5.4 Capturas de cada prueba:**

### **5.4.1 Prueba 1:**



******5.4.2 Prueba 2:**

**5.4.3 Prueba 3:**



**5.5 Implementación:**

#include <stdio.h> //Librerias, se incluyen las funciones estandar de entrada y salida

#include <windows.h>//Libreria, de la cual usaremos la función Sleep

//Programa principal

int main**(**void**){**

//Declaración de variables, n es el número de veces que repetiremos

//el ciclo, primero y segundo guardan números, que serán sumados

//elemento es el número de veces que hemos repetido el ciclo, un contador

//siguiente variable donde almacenaremos la suma de primero más segundo

int n**,** primero**,** segundo**,** elemento**,** siguiente**;**

//Inicio del ciclo de repetición

**do{**

//Mandamos un mensaje al usuario

printf**(**"Ingresa un numero -> "**);**

//Lectura y almacenamiento de datos en la variable n

scanf**(**"%d"**,** **&**n**);**

//Mientras n no sea mayor a 2 se seguirá repitiendo el ciclo

**}while(!(**n**>**2**));**

//Igualamos la variable primero a 0

primero **=** 0**;**

//Mostramos el valor que tiene l variable primero

printf**(**"%d "**,** primero**);**

//Esperamos 0.5 segundos

Sleep**(**500**);**

//Segundo lo igualamos a 1

segundo **=** 1**;**

//Mostramos el valor de segundo

printf**(**"%d "**,** segundo**);**

//Esperamos 0.5 segundos

Sleep**(**500**);**

//Variable elemento la igualamos a 3

elemento **=** 3**;**

//Inicio del ciclo de repetición

**do{**

//Variable siguiente almacenamos la sumatoria

//entre variables primero y segundo

siguiente **=** primero **+** segundo**;**

//Mostramos el valor que tiene la variable siguiente

printf**(**"%d "**,** siguiente**);**

//Esperamos 0.5 segundos

Sleep**(**500**);**

//La variable primero la igualamos con segundo

primero **=** segundo**;**

//La variable segundo la igualamos con siguiente

segundo **=** siguiente**;**

//Hacemos un conteo usando la variable elemento

//sería lo mismo si tenemos elemento++

elemento **=** elemento **+** 1**;**

//Este ciclo se repite mientras elemento no sea mayor a la variable n

**}while(!(**elemento**>**n**));**

**return** 0**;**

**}**