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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ing. Claudia Rodríguez Espino |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | 1102 |
| *No de Práctica(s):* | 07 |
| *Integrante(s):* | Alejandro Meneses Mercado |
|  |  |
|  |  |
| *Semestre:* | 2018-1 |
| *Fecha de entrega:* | 29 de septiembre del 2017 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Práctica 07: Fundamentos de Lenguaje C

Objetivo:

Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

Desarrollo:

Primero vimos un poco de historia del lenguaje de programación C, el cual ofrece ventajas como economía de expresión, control de flujo y estructuras de datos y un conjunto de operadores.

También vimos que para crear un programa en C se siguen tres etapas principales: edición, compilación y ejecución.

En la edición: Se escribe el código fuente en lenguaje C desde algún editor de textos.

En la compilación: A partir del código fuente se se crea el ejecutable.

En la ejecución: El ejecutable se puede ejecutar en la arquitectura correspondiente.

Algo muy importante que vimos fue que el lenguaje de programación consiste en una o más funciones, donde una de ellas debe llamarse main() y es la principal.

Vimos que esta función principal puede contener sentencias, estructuras de control y comentarios, aparte, se encuentran la declaración y/o asignación de variables, la realización de operaciones básicas, y las llamadas a funciones.

Primero vimos como se incluyen los comentarios por línea y por bloque .

El comentario por línea inicia cuando se insertan los símbolos '//' y termina con el salto de línea y el comentario por bloque inicia cuando se insertan los símbolos '/\*' y termina cuando se encuentran los símbolos '\*/'.

Vimos que hacer comentarios es una buena práctica para ocumentar el programa.

Luego vimos la declaración de variables en donde sus sintaxis es: [modificadores] tipoDeDato identificador [= valor]; donde los modificadores son opcionales, se debe declarar el tipo de dato que puede contener la variable, debe declarar el identificador con el que se va a manejar el valor y se puede asignar un valor inicial a la variable, esto es opcional.

Posteriormente vimos los tipos de datos básicos en C, los cuales son:

Caracteres: codificación definida por la máquina; Enteros: números sin punto decimal; Flotantes: números reales de precisión normal; Dobles: números reales de doble precisión.

Después vimos las variables en enteras que existen en C, las cuales son: signed char, unsigned char, signed short, unsigned short, signed int unsigned int, signed long, unsigned long, enum.

Aparte vimos las variables reales que existen en C, las cuales son: float, double y long double.

También vimos que para poder acceder al valor de una variable se requiere especificar el tipo de dato. Los especificadores que tiene lenguaje C para los diferentes tipos de datos son:

Entero %d, %i, %ld, %li, %o, %x, Flotante %f, %lf, %e, %g, Carácter %c, %d, %i, %o, %x, Cadena de caracteres %s.

Y vimos como utilizar estos especificadores de conversión dependiendo de lo que queramos.

Posteriormente vimos lo que son es un identificador, el cual es el nombre con el que se va a almacenar en memoria un tipo de dato. Los identificadores deben seguir 2 reglas; Debe iniciar con una letra [a-z] y puede contener letras [A-Z, a-z], números [0-9] y el carácter guión bajo (\_).

Después vimos que la biblioteca 'stdio.h' contiene diversas funciones tanto para imprimir en la salida estándar como para leer de la entrada estándar.

printf es una función para imprimir con formato, es decir, se tiene que especificar entre comillas el tipo de dato que se desea imprimir, también se puede combinar la impresión de un texto predeterminado.

scanf es una función que sirve para leer datos de la entrada estándar, para ello únicamente se especifica el tipo de dato que se desea leer entre comillas y en qué variable se quiere almacenar; es importante saber que se debe utilizar este símbolo & para guardar el dato en la memoria.

Otra cosa que vimos fueron los modificadores de alcance, los cuales se pueden agregar al inicio de la declaración de variables y son const y static.

El modificador const impide que una variable cambie su valor durante la ejecución del programa, es decir, permite crear constantes.

El modificador static indica que la variable permanece en memoria desde su creación y durante toda la ejecución del programa, es decir, permanece estática en la memoria.

Aparte vimos cuales eran los operadores aritméticos que maneja el lenguaje C:

Suma +, Resta -, Multiplicación \*, División / y Módulo %.

Y Vimos los operadores lógicos utilizados en C:

>> Corrimiento a la derecha, << Corrimiento a la izquierda, & Operador AND, |Operador OR y ~ Complemento ar-1.

Después vimos que el resultado de una operación entre dos tipos de datos iguales puede dar como resultado un tipo de dato diferente, en esos casos es necesario moldear el resultado. A este proceso se le conoce como cast.

Por último vimos que las expresiones lógicas están constituidas por números, caracteres, constantes o variables que están relacionados entre sí por operadores lógicos. Una expresión lógica puede tomar únicamente los valores verdadero o falso.

Y son los sig: == Igual que, != Diferente a, < Menor que, > Mayor que, < = Menor o igual, > = Mayor o igual.

También vimos que los operadores lógicos permiten formular condiciones complejas a partir de condiciones simples: ! No, && Y, || O.

Y vimos como hacer incremetos y decrementos, en los incrementos se puede usar ++n para un preincremento y n++ para un posincremeto.

Se usa los mismo para el decremento solo cambia el - -n y n- -.

Como actividad tuvimos que hacer 3 programas, de los cuales 2 de ellos debían cumplir ciertas condiciones.

1. Programa que suma 2 numeros.

Código:

#include<stdio.h>

float a,b,c;

main()

{

printf("\n\t\tPrograma que realiza la suma de 2 numeros\n");

printf("\nDame el valor de tu primer numero: ");

scanf("%f",&a);

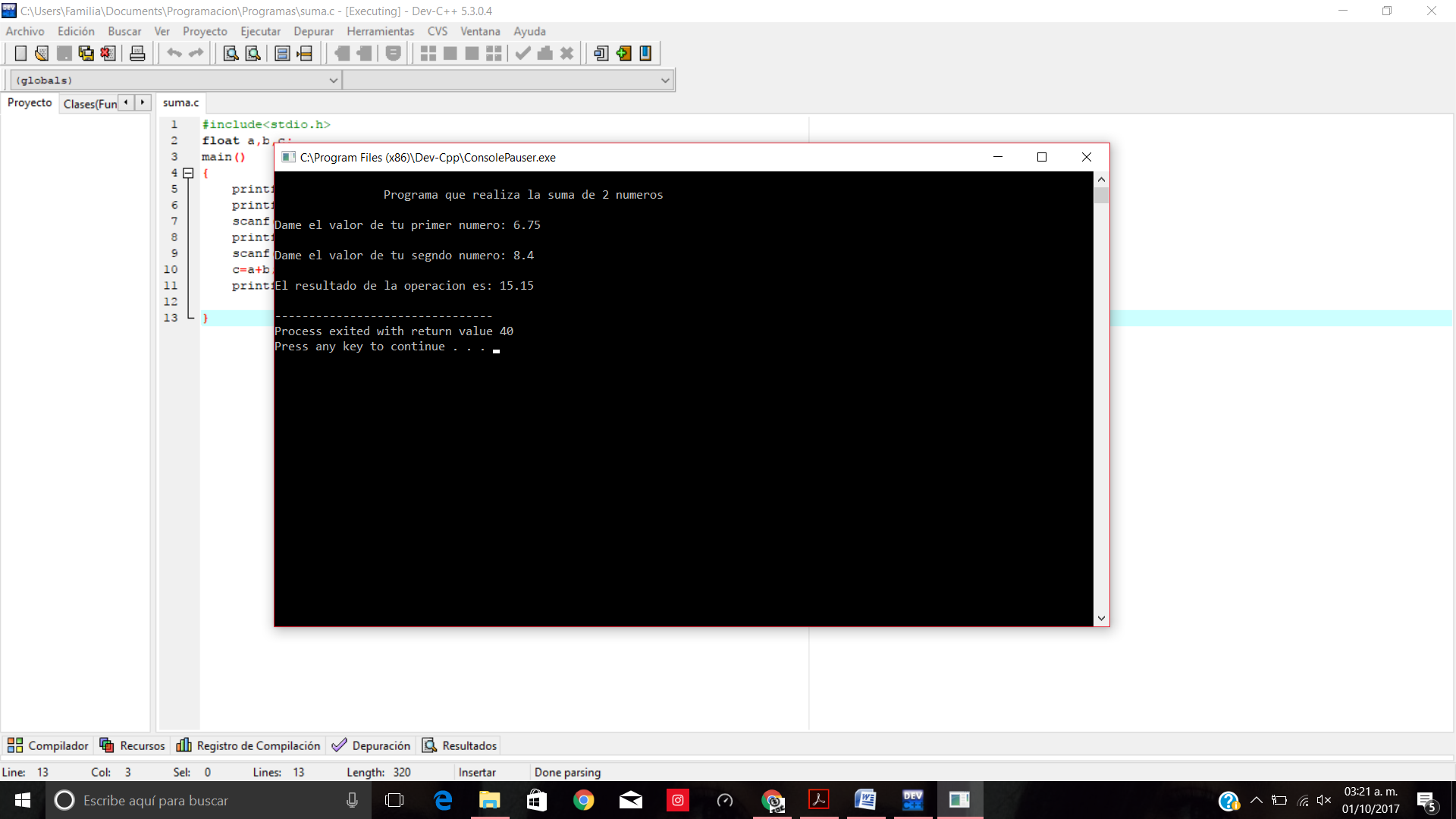
printf("\nDame el valor de tu segndo numero: ");

scanf("%f",&b);

c=a+b;

printf("\nEl resultado de la operacion es: %.2f\n", c);

}



1. Programa que calcula el area de un círculo(usando const PI)

Código:

#include<stdio.h>

const float PI = 3.1416;

float r,a;

main()

{

printf("\n\t\tPrograma que calcula el area de un circulo\n");

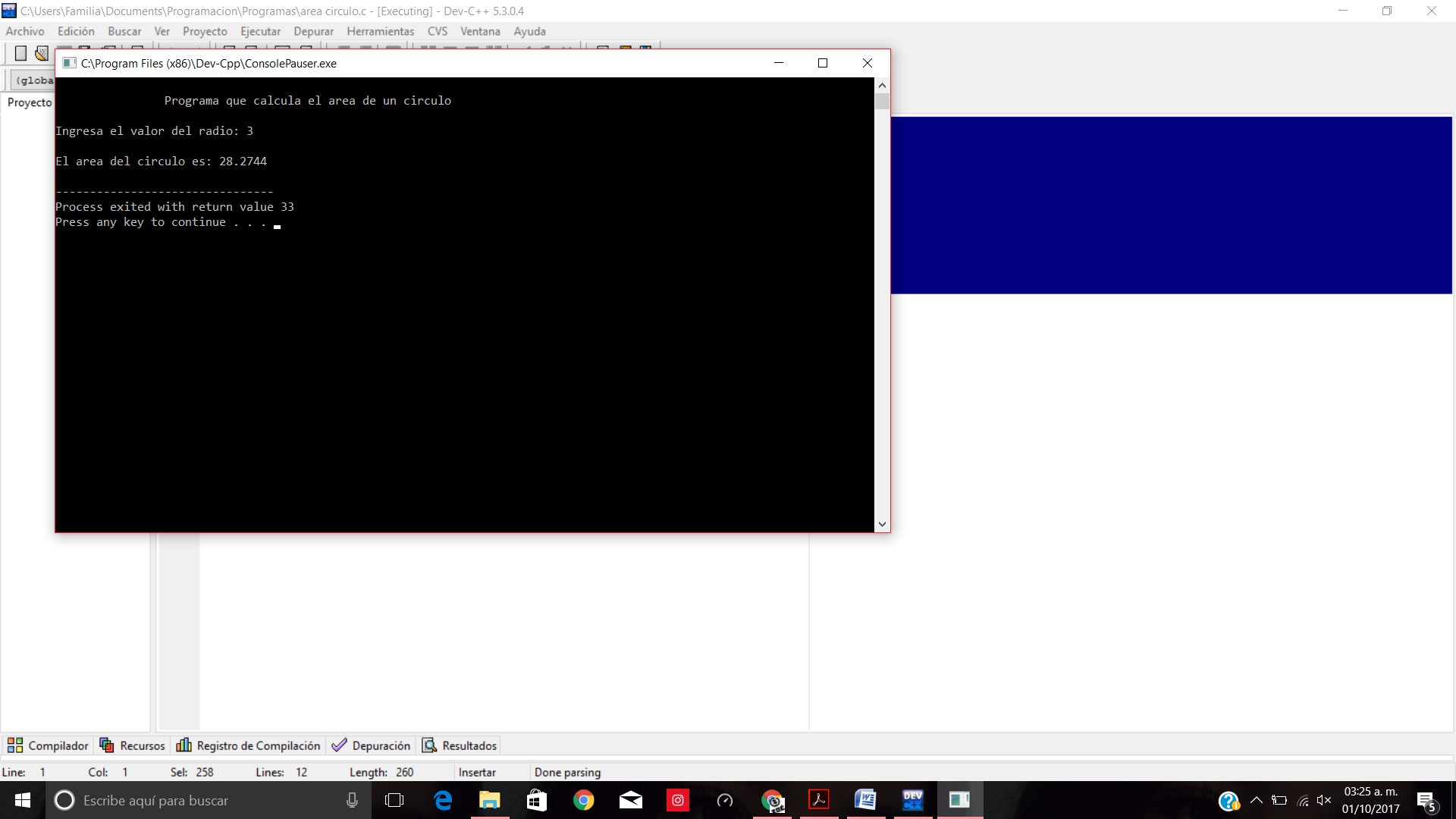
printf("\nIngresa el valor del radio: ");

scanf("%f",&r);

a=PI\*r\*r;

printf("\nEl area del circulo es: %.4f\n", a);

}



1. Programa que calcula el promedio de 3 números (haciendo el moldeo o cast)

Código:

#include <stdio.h>

int a=1;

float p,cont=0,num;

main()

{

printf("\n\t\tPrograma que calcula el promedio de 3 numeros\n");

while(a<4)

{

printf("\nDame el valor %d: ",a);

scanf("%f",&num);

cont=(cont+num);

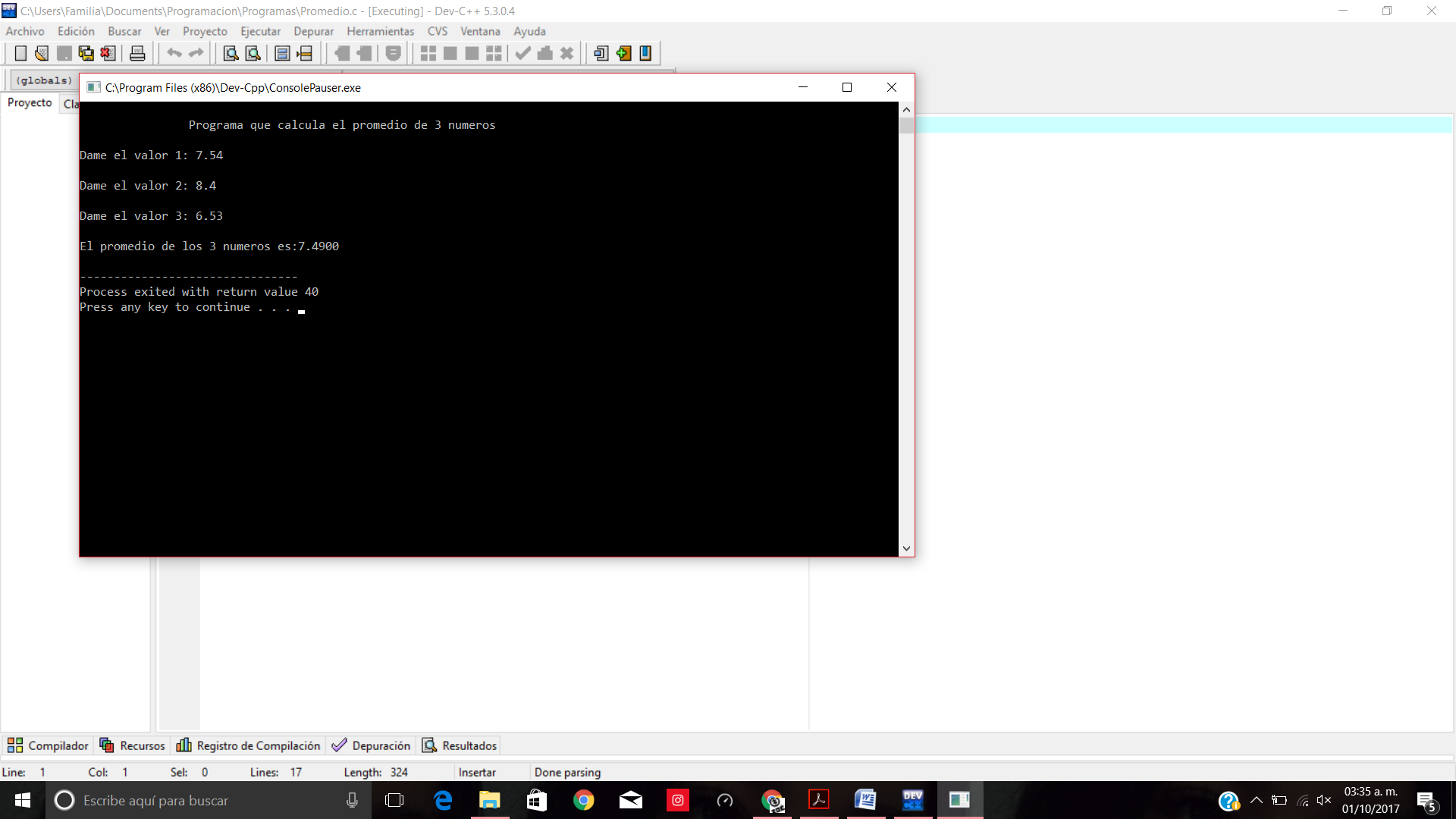
a++;

}

double p=(double)cont/3;

printf("\nEl promedio de los 3 numeros es:%.4f\n",p);

}



Conclusiones:

En esta práctica recordamos algunos conceptos ya vistos en clase, los cuales nos ayudaron a entender mejor esta práctica. También nos servira para todas las prácticas que vienen puesto que aqui vimos todas las bases del lenguaje de Programación C.