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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
|  | RODRIGUEZ ESPINO CLAUDIA |
| *Asignatura:* | FUNDAMENTOS DE PROGRAMACIÓN |
| *Grupo:* | 1104 |
| *No de Práctica(s):* | PRÁCTICA 10 |
| *Integrante(s):* | VALENCIA MEDINA ISAC |
|  |  |
|  |  |
| *Semestre:* | PRIMER SEMESTRE 2019-1 |
| *Fecha de entrega:* | 15/OCTUBRE/2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

GUÍA PRÁCTICA DE ESTUDIO 10: Depuración de programas

**OBJETIVO:**

Aprender las técnicas básicas de depuración de programas en C para revisar de manera precisa el flujo de ejecución de un programa y el valor de las variables; en su caso, corregir posibles errores.

**ACTIVIDADES:**

♣ Revisar, a través de un depurador, los valores que va tomando una variable en un programa escrito en C, al momento de ejecutarse.

♣ Utilizando un depurador, revisar el flujo de instrucciones que se están ejecutando en un programa en C, cuando el flujo depende de los datos de entrada.

**INTRODUCCIÓN:**

Depurar un programa significa someterlo a un ambiente de ejecución controlado por medio de herramientas dedicadas a ello. Este ambiente permite conocer exactamente el flujo de ejecución del programa, el valor que las variables adquieren, la pila de llamadas a funciones, entre otros aspectos. Es importante poder compilar el programa sin errores antes de depurarlo. Antes de continuar, es necesario conocer las siguientes definiciones (extraídas del Glosario IEEE610) ya que son parte latente del proceso de Desarrollo de Software: Error. Se refiere a una acción humana que produce o genera un resultado incorrecto. Defecto (Fault). Es la manifestación de un error en el software. Un defecto es encontrado porque causa una Falla (failure). Falla (failure). Es una desviación del servicio o resultado esperado.

**Problema 1: Fórmula General**

**#include <stdio.h>**

**#include <conio.h>**

**#include <math.h>**

**int a,b,c,d;**

**float x1,x2,r,m;**

**main(){**

**printf("Problema 9:F%cmula General\n",162);**

**printf("Dame el valor de a: \n");**

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

**printf("Dame el valor de b: \n");**

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

**printf("Dame el valor de c: \n");**

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

**if(a>0){**

**d=2\*a;**

**m=(b\*(-1));**

**m=m/d;**

**r=((b\*b)-(4\*a\*c));**

**if(r>=0){**

**r=(sqrt(r)/d);**

**x1=m+r;**

**x2=m-r;**

**printf("x1 es igual a: %.2f\n",x1);**

**printf("x2 es igual a: %.2f",x2);**

**}else{**

**r=r\*(-1);**

**r=(sqrt(r)/d);**

**printf("x1 es igual %.2f + %.2f i\n",m,r);**

**printf("x2 es igual %.2f - %.2f i",m,r);**

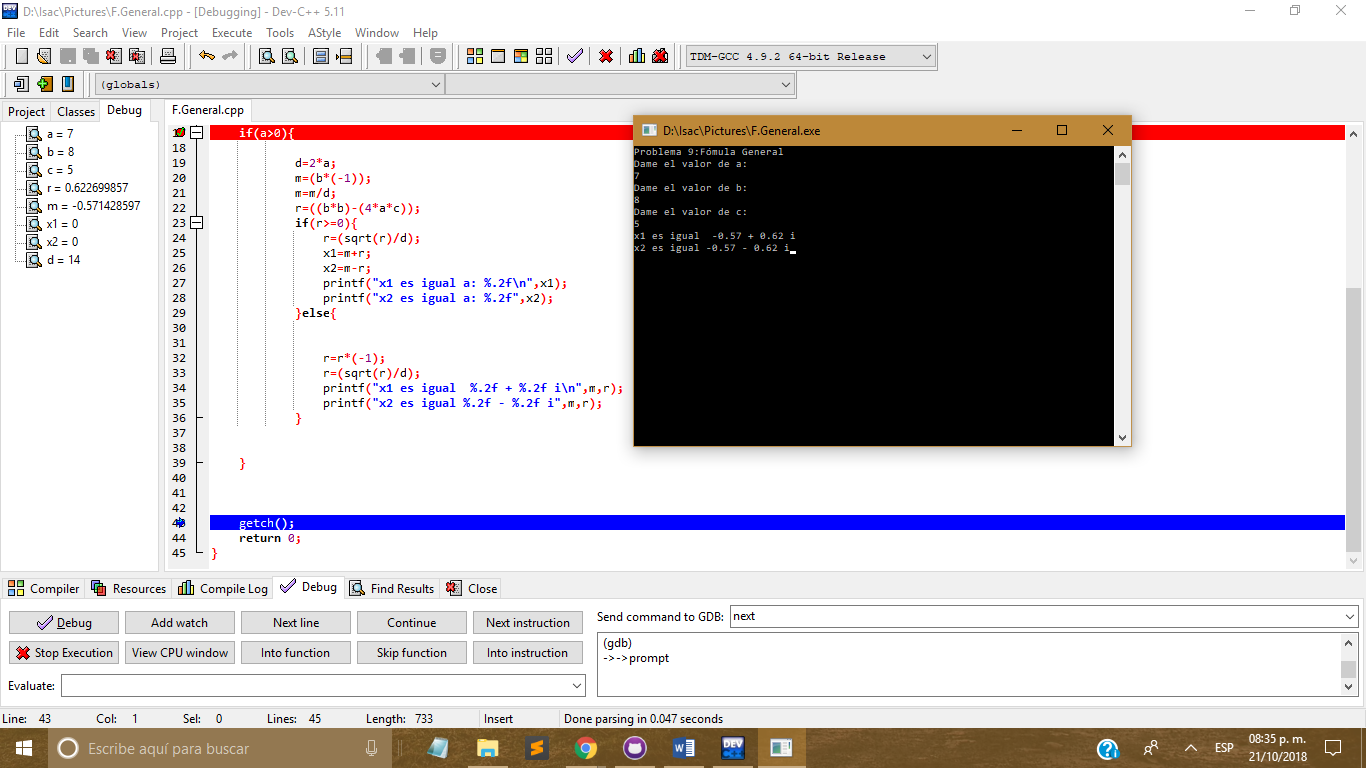
**}**

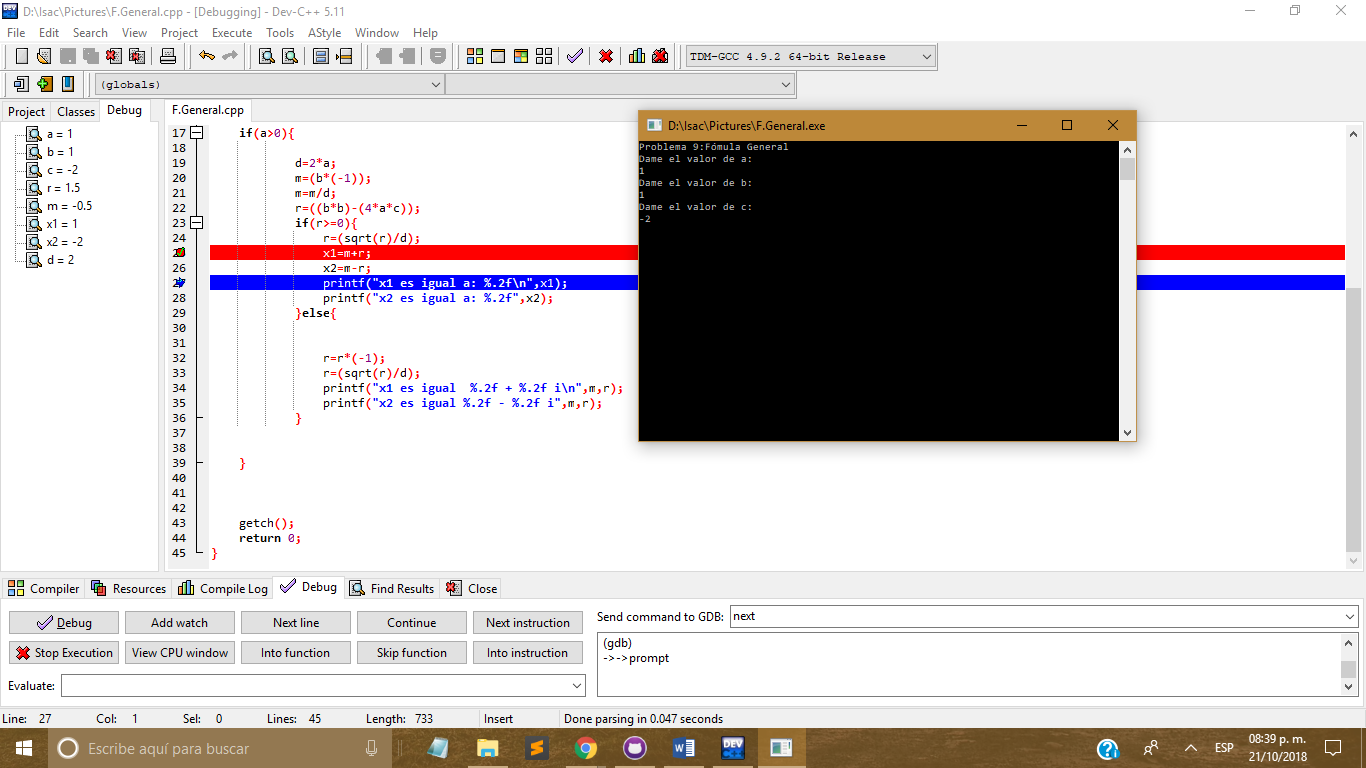
**}**

**getch();**

**return 0;**

**}**





**Problema 2: Sistema de Ecuaciones**

**#include <stdio.h>**

**#include <conio.h>**

**int x,y;**

**main(){**

**printf("Problema 8: Resoluci%cn de ecuaciones\n",162);**

**printf("Dame un valor para 'x': ");**

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

**if(x>2){**

**printf("Ecuaci%cn para x>2: y=x^2+3x-2\n",162);**

**y=x\*x+3\*x-2;**

**printf("Valor de y: %d",y);**

**}else if(x<2){**

**printf("Ecuaci%cn para x<2: y=2x^2+x+8\n",162);**

**y=2\*x\*x+x+8;**

**printf("Valor de y: %d",y);**

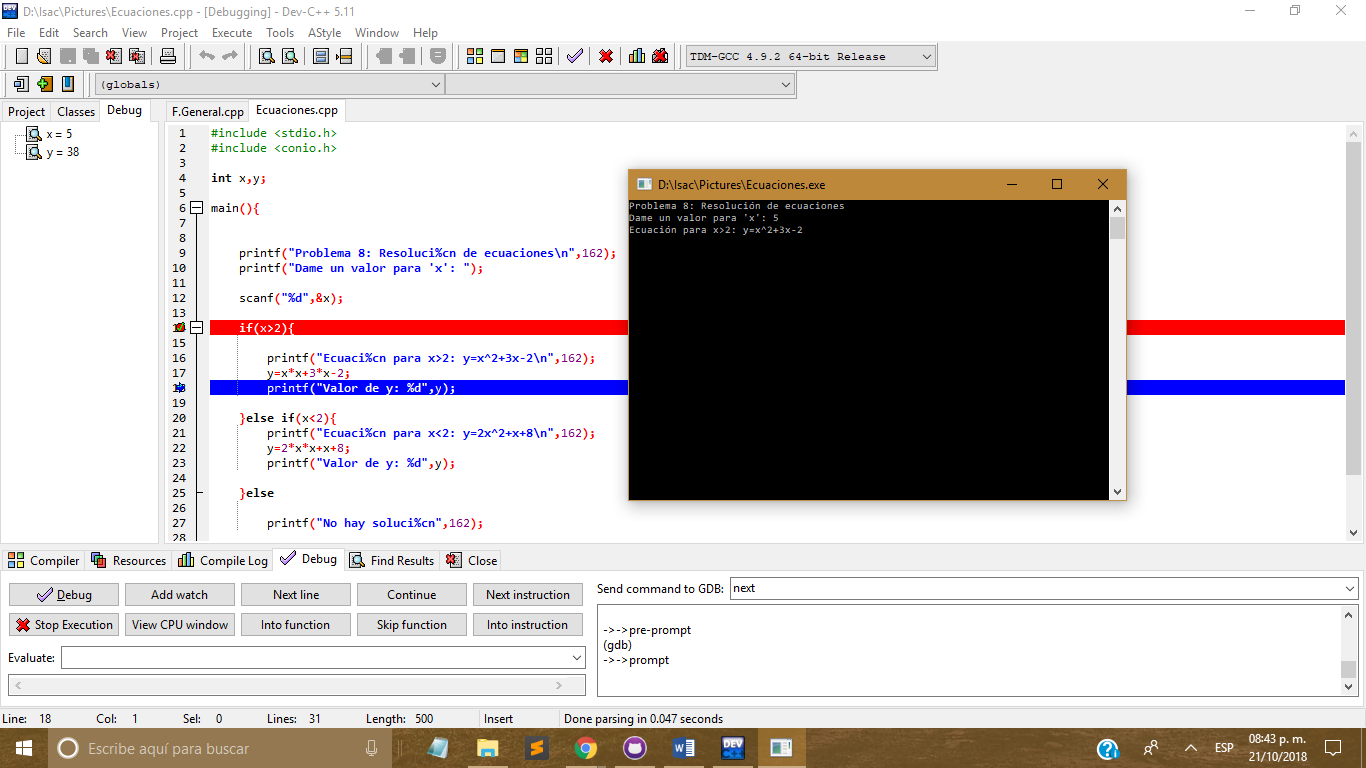
**}else**

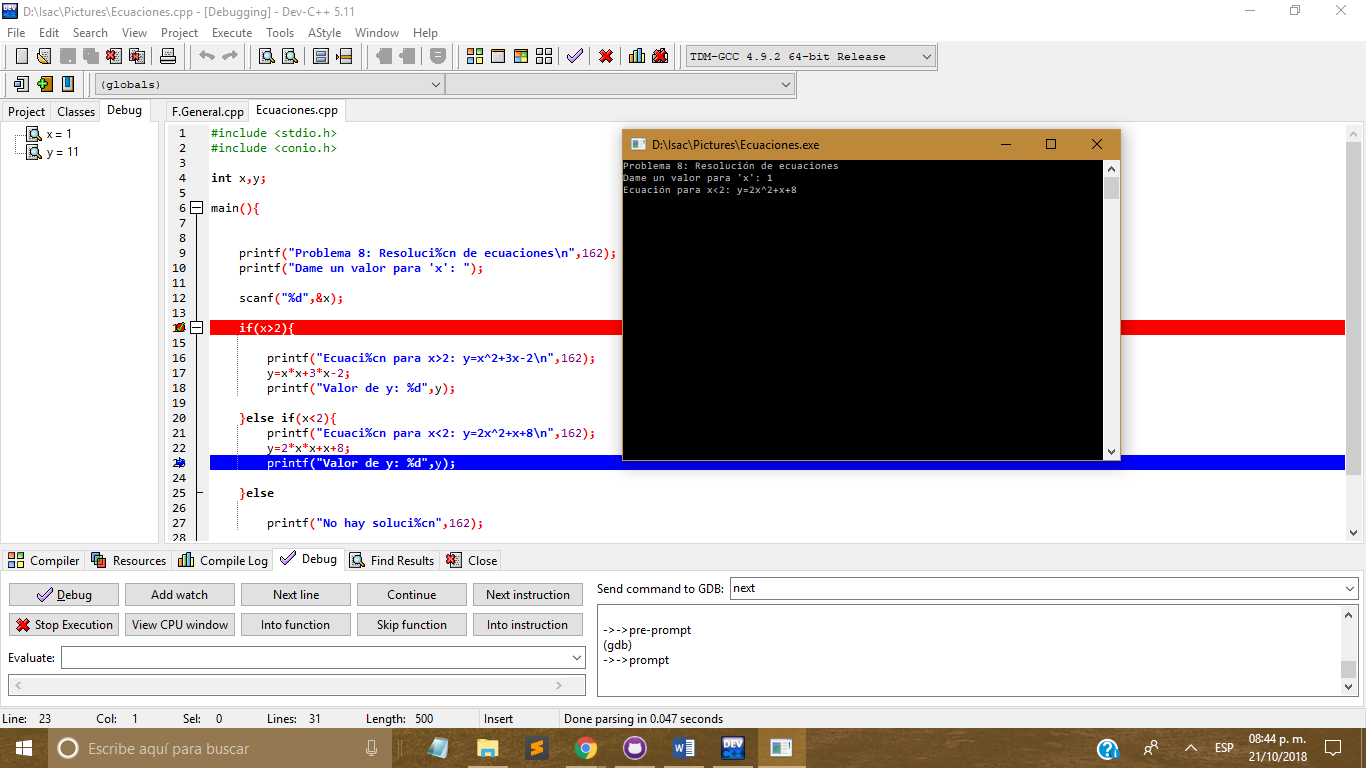
**printf("No hay soluci%cn",162);**

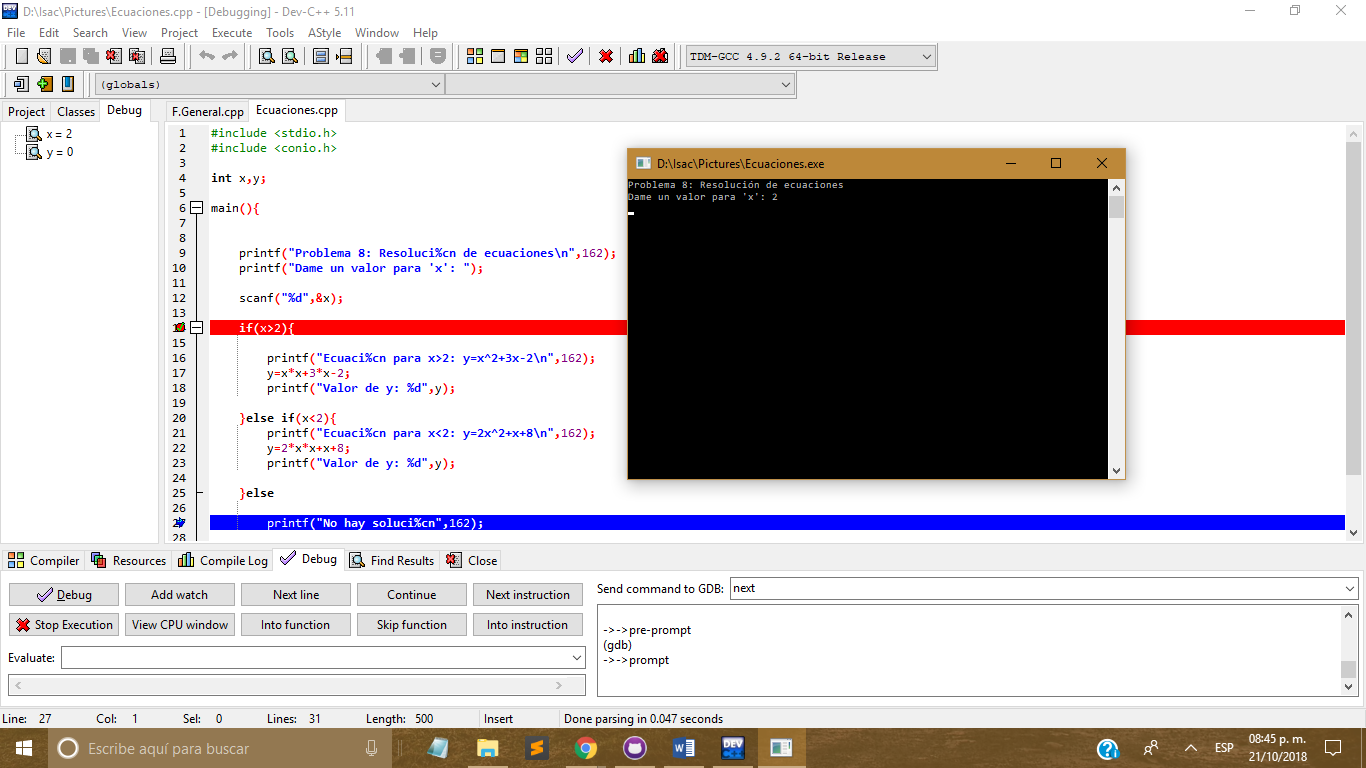
**getch();**

**return 0;**

**}**







**Problema 3: Gastos Mensuales**

**#include <stdio.h>**

**#include <conio.h>**

**float gastos[15];**

**float suma=0;**

**float promedio;**

**int x;**

**main(){**

**printf("Problema 18: Gastos Mensuales de una a%co y promedio anual con arreglos\n",164);**

**for(x=1;x<13;x++){**

**printf("Dame el Gasto del mes [%d]: ",x);**

**scanf("%f",&gastos[x]);**

**suma=(suma+gastos[x]);**

**}**

**printf("\nGastos almacenados:\n");**

**for(x=1;x<13;x++){**

**printf("Los gastos alamacenados en el mes [%d] son: %.2f\n",x,gastos[x]);**

**}**

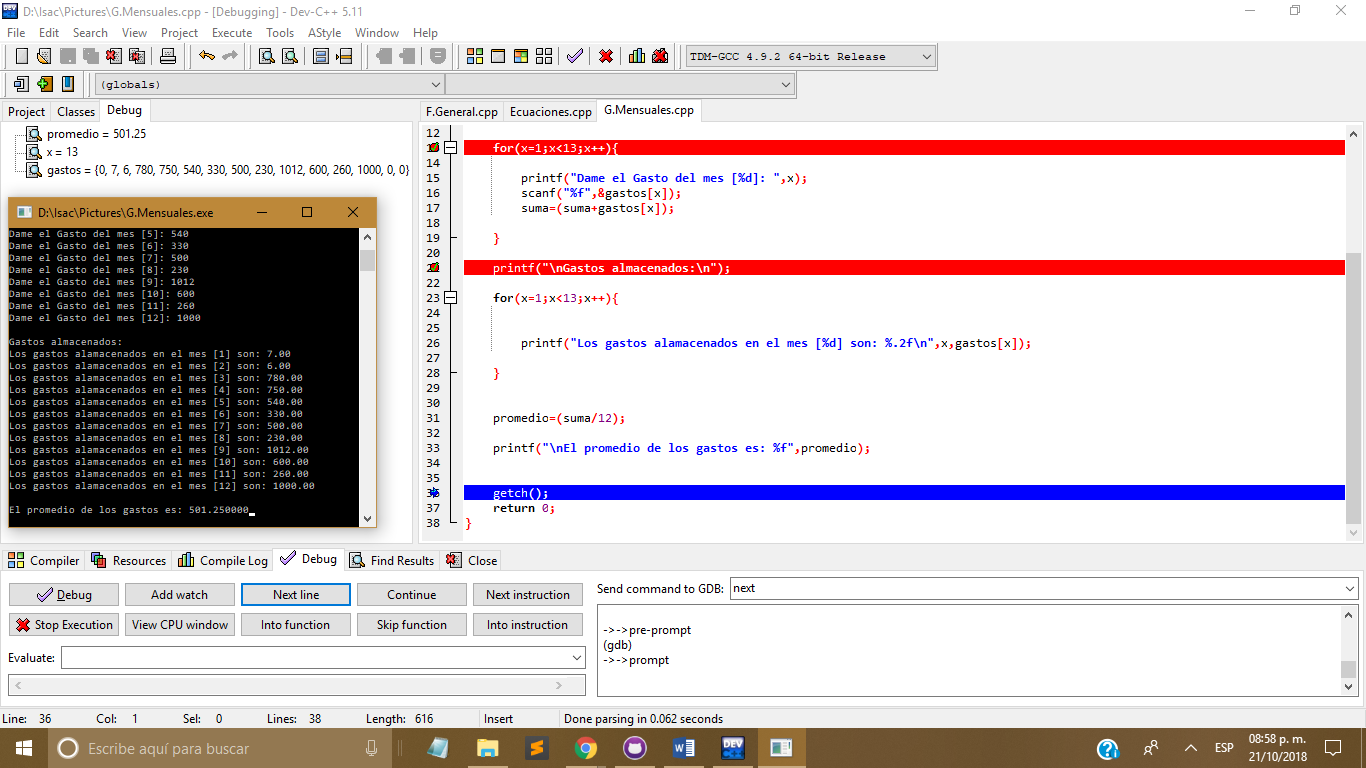
**promedio=(suma/12);**

**printf("\nEl promedio de los gastos es: %f",promedio);**

**getch();**

**return 0;**

**}**



**CONCLUSIÓN:**

La herramienta de depuración que nos brindan los distintos IDE (Entornos de Desarrollo Integrado) o bien compiladores, es sumamente útil si queremos ver a detalle lo que sucede durante la ejecución de nuestro programa. Hay muchas ocasiones en donde tenemos declaradas variables que a lo largo de la ejecución y de forma dinámica cambiaran sus valores hasta que se termine la ejecución, en ocasiones suele suceder que al final el resultado no es el esperado, es entonces cuando revisamos el código y analizamos si algo está mal declarado o hay error de sintaxis pero si no detectamos nada seguramente el problema se encuentra en el valor que están tomando las variables, es en este momento cuando la herramienta de depuración se vuelve primordial. Al utilizar la depuración nos damos cuenta que es lo que pasa línea por línea y cómo van cambiando los valores de nuestra variables, entonces en el momento en que detectamos que el valor de alguna variable es incorrecto, ya tenemos bien identificado en donde empezó a fallar y podemos corregirlo, me ha pasado que después de la ejecución de mi programa me devuelve cero de resultado y no debería de ser, así que comienzo a checar como cambian las variables y lo más común es que las variables declaradas son de distinto tipo y por eso no guardan el valor o que simplemente su rango de tamaño es insuficiente y no se guarda nada. Para esta práctica aprendimos como utilizar esta herramienta en Dev c++, en lo personal nunca la había utilizado y considero que para problemas sobre todo que involucren ciclos es muy útil.