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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Claudia Rodriguez Espino |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 1104 |
| *No de Práctica(s):* | 10 |
| *Integrante(s):* | Aguilera Valderrama Alexis Fernando |
|  |  |
| *No. de Equipo de cómputo empleado* | 1 |
| *Semestre:* | 1 |
| *Fecha de entrega:* | 21/10/2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Objetivos:

* 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.

Desarrollo:

Se revisó el concepto de depuración de un programa, el cual dice que se ejecuta en un ambiente controlado el programa a codificar para ver su comportamiento con más detenimiento y encontrar anomalías.

La depuración es útil cuando:

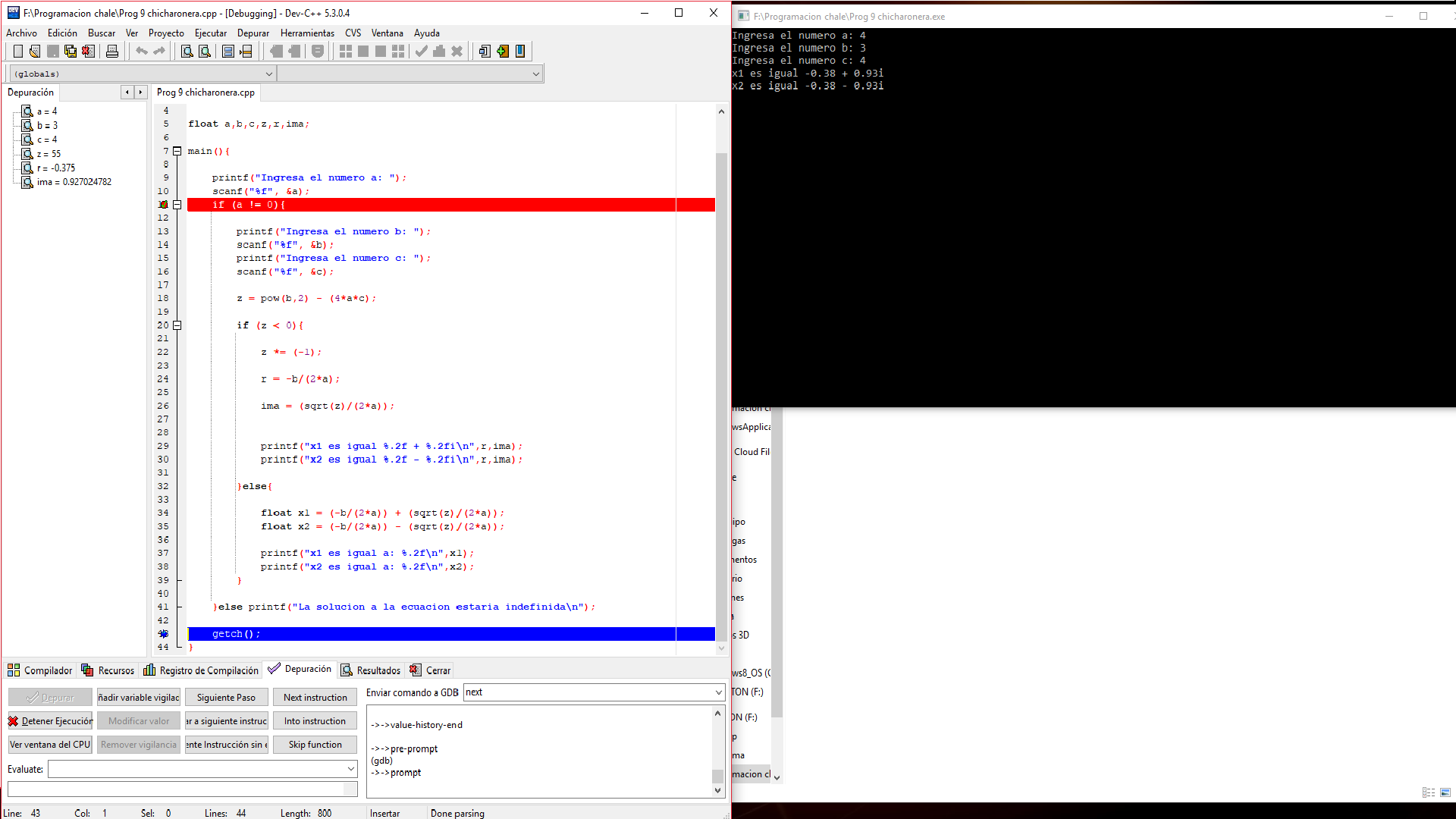
* Se desea optimizar el programa
* El programa tiene algún fallo
* Tiene algún error de ejecución o defecto

Algunas herramientas que nos ofrece la depuración son:

* Ejecutar el programa
* Mostrar el código fuente
* Punto de ruptura
* Continuar
* Ejecutar el programa de nuevo
* Ejecutar la siguiente instrucción
* Visualizar el valor de las variables

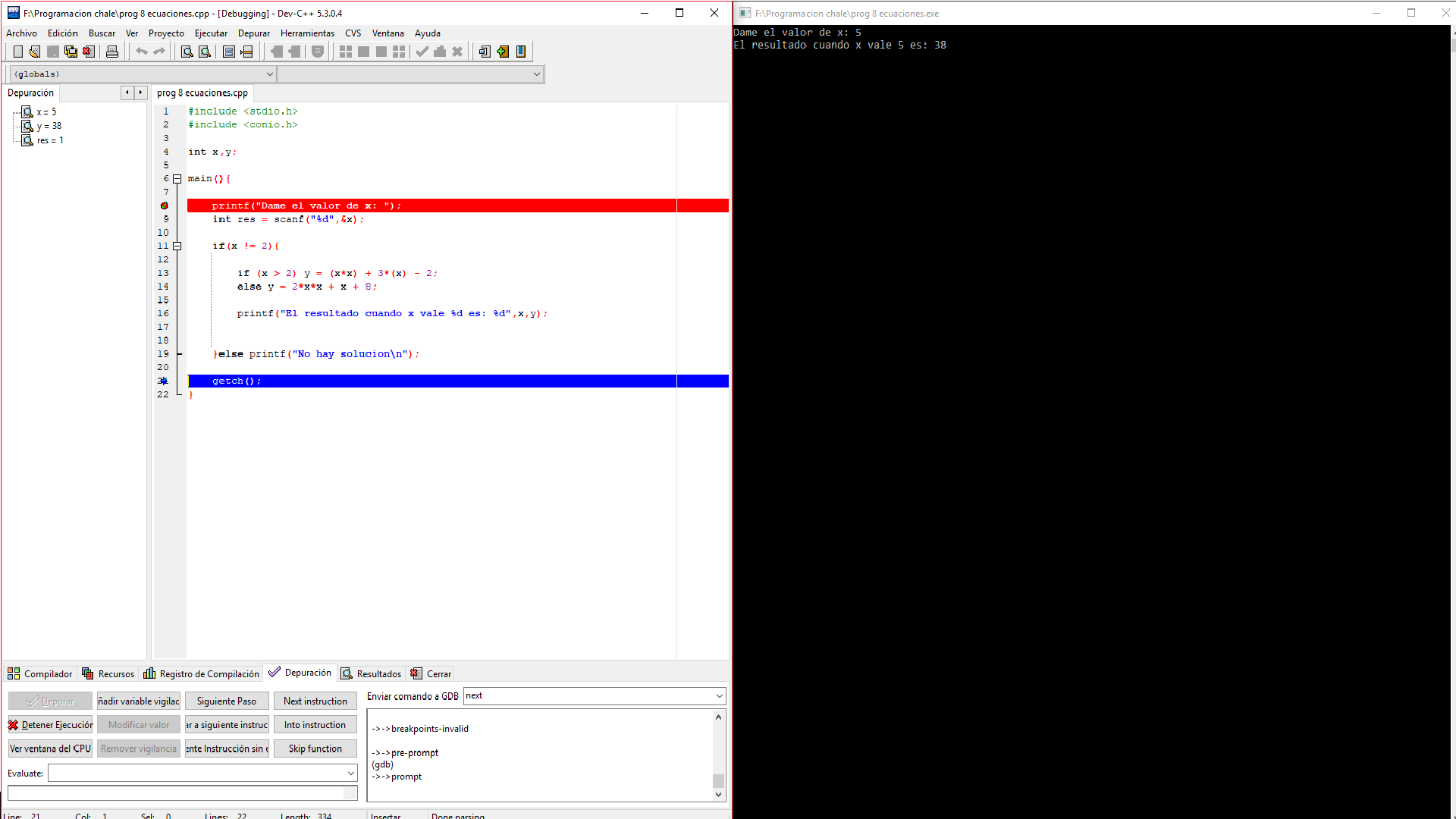
Para esta práctica se utilizó la herramienta de depuración integrada en DevC++ y pusieron en depuración los siguientes programas:

Resolución de ecuaciones de 2do grado.

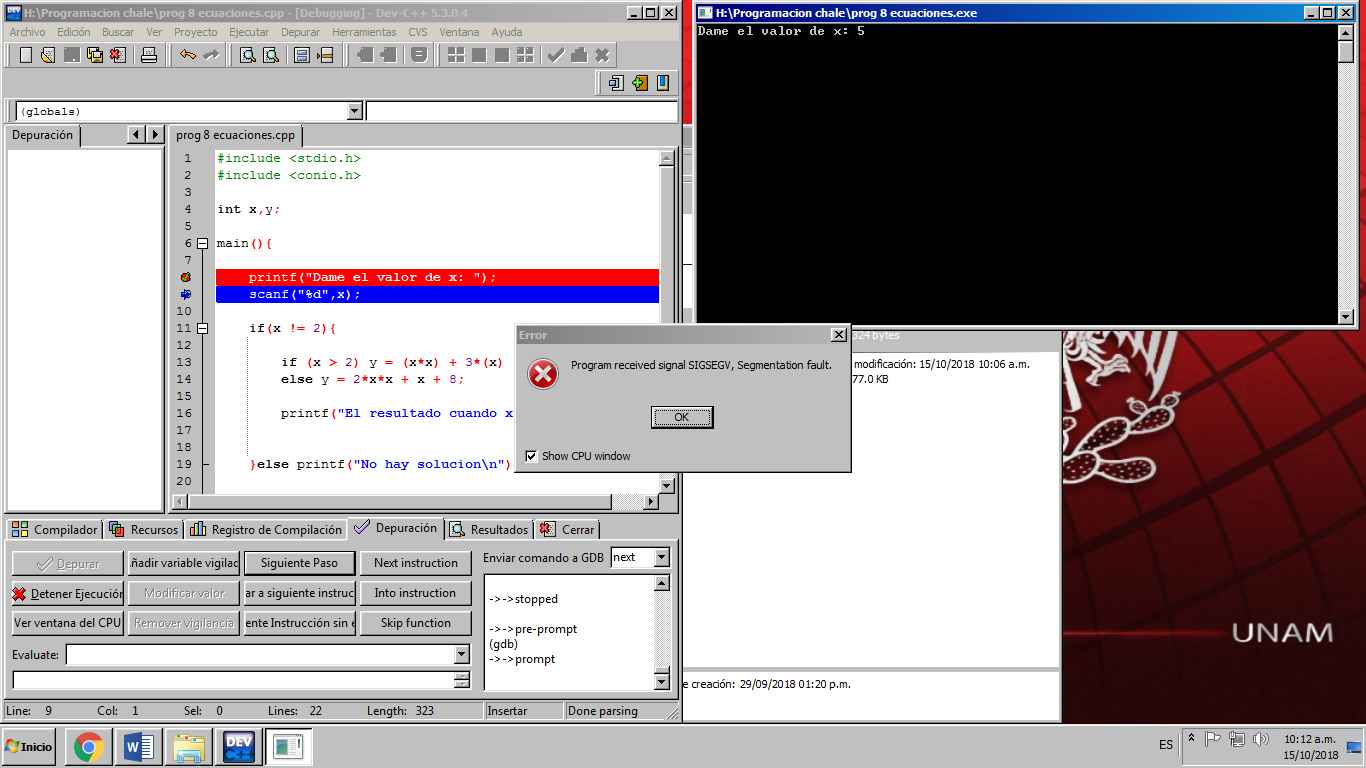


En el ejemplo anterior se puede apreciar que se añadieron vigilancias para las variables y un punto de ruptura en la línea número 11 para ir viendo el comportamiento del programa línea por línea desde ese punto.

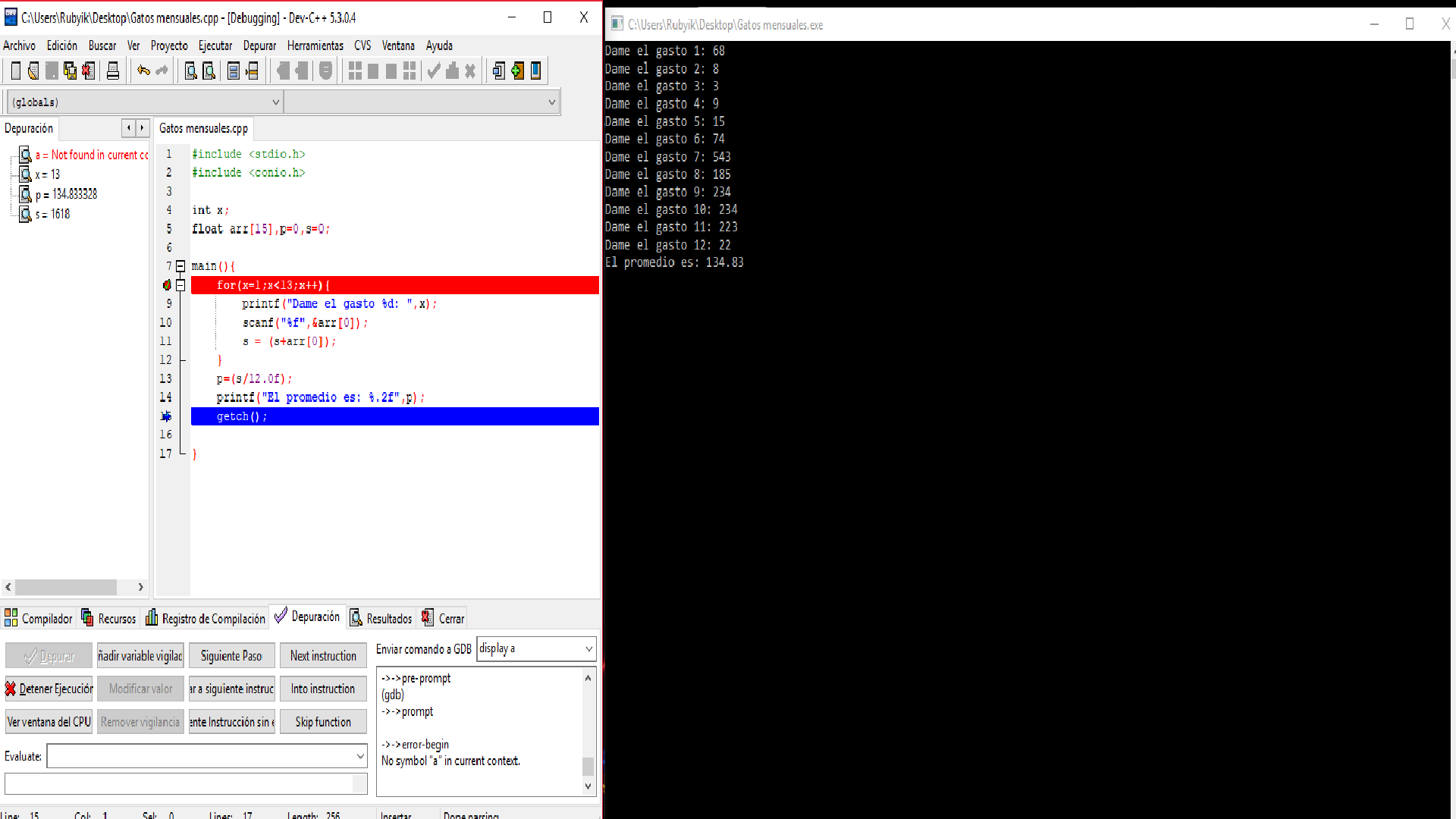
Se hizo lo mismo con el programa de funciones con dos reglas de correspondencia.



Para probar las funcionalidades que nos ofrece el depurador de DevC++, del programa anterior se quitó el “&” de la x para forzar un error, el resultado fue que cuando se intentó correr esa línea de código, aparte de que el programa dejó de funcionar, el depurador dio un error de violación de segmento.



Gastos mensuales con arreglos.



Conclusión:

Se logró usar la herramienta de depuración para encontrar errores y maneras de mejorar el rendimiento del programa.

Las herramientas más útiles que podemos utilizar para depurar un programa es el inspector de variables y el punto de ruptura para ejecutar la siguiente línea cuando nosotros lo queramos.