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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Rodriguez Espino Claudia |
| *Asignatura:* | Fundamentos de Programacion |
| *Grupo:* | 1102 |
| *No de Práctica(s):* | Practica numero 10 |
| *Integrante(s):* | Gutierrez Orozpe Luis Fernando |
|  |  |
|  |  |
| *Semestre:* | 2018-1 |
| *Fecha de entrega:* | 12 de Noviembre de 2017 |
| *Obervaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

OBJETIVOS

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

Para la ejecución de esta practica se nos pide que depuremos 3 programas, 2 que vienen incluidos en la practica y un tercero que consiste en la formula general (chicharronera) cuya función es ejecutar la formula cuadrática para poder obtener las raíces de una ecuación de segundo grado.

**DEPURACION**

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 de las variables que adquieren, la pila de llamadas a funciones, entre otros aspectos. Es importante poder compilar el programa sin errores antes de depurarlo.

**DEPURACION DE PROGRAMAS ESCRITOS EN C CON DEV-C++**

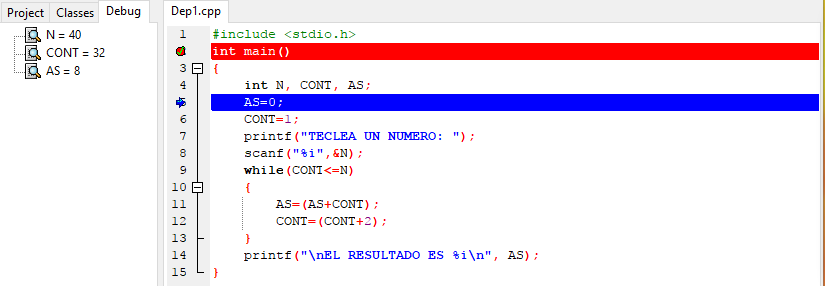
Para que se pueda llevar a cabo la depuración en Dev-C++ es necesario activar la opción de “Generar información de depuración” en las herramientas del programa. Después de realizar lo anterior, el programa realizado puede compilarse y ejecutarse con lo que ofrece el menú Ejecutar.

Para depurar el programa, primero se debe compilar con el menú Ejecutar y luego depurar con el menú Depurar. El programa se abrirá y se ejecutará hasta el primer punto de ruptura señalado. También se abrirá un cuadro de herramientas en la parte inferior del programa que tiene las principales herramientas de depuración en la parte derecha.

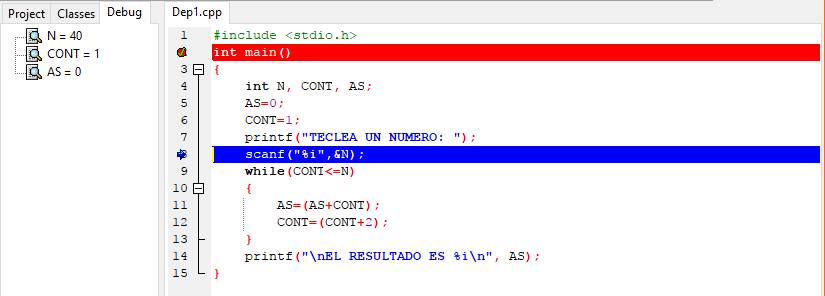
Para detener la depuración puede seleccionarse la opción “parar ejecución”. Para estudiar el valor de cada variable, se puede recurrir a la función “Añadir Watch” y escribir el nombre de la variable. En un cuadro a la izquierda, se verá el nombre de la variable y su valor hasta el punto donde se está ejecutando el programa.

A continuación se anexan las capturas de pantalla de los programas que se van a depurar para poder entender su funcionamiento y si es el caso, cuáles son sus errores.

**PROGRAMA 1: ANEXADO A LA PRACTICA**

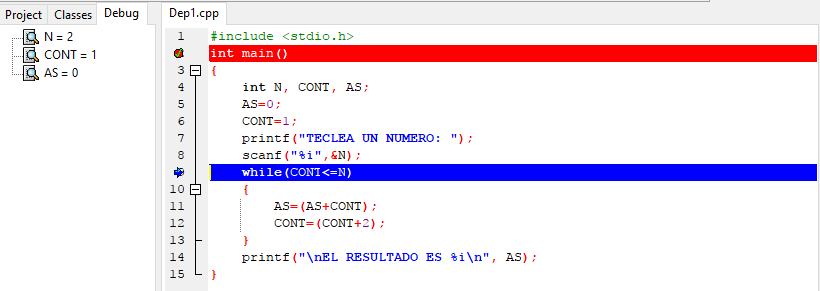


En este punto de la depuración lee las variables y les asigna valores al azar ya que no se ha especificado que valores debe tomar cada variable, en este punto en el ejecutable aún no se muestra nada en pantalla.



En este punto de la depuración los valores de las variables AS y CONT cambiaron debido a que el programa especifica que AS debe ser igual a 0 y CONT debe ser igual a 1, de la misma forma, en el ejecutable ya se muestra “Teclea un numero” debido al printf.

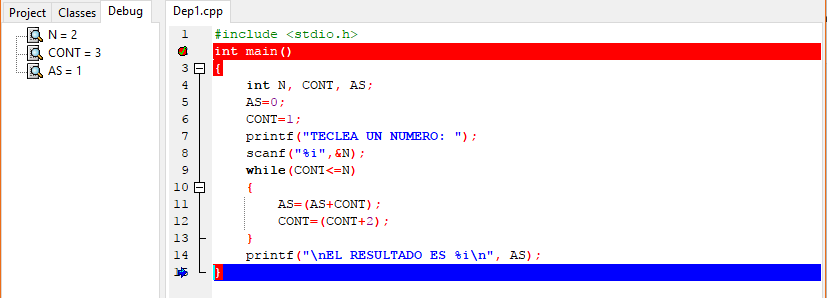




En este punto de la depuración nos pide ingresar un valor a la variable N a lo que se ingresa el valor de 2, por lo que el programa entra a el ciclo while. Al realizar las operaciones tenemos que:

AS=(0+1) lo que da como resultado 1, después se ejecuta la operación CONT=(1+2) que nos da como resultado que CONT ahora es igual al número 3.

Al terminar este bloque de instrucciones regresa al ciclo while en el que analiza si el valor de CONT es menor que el valor de la variable N, como CONT vale 3 y la variable N vale 2 entonces no entra al ciclo while y continua con el código fuente.

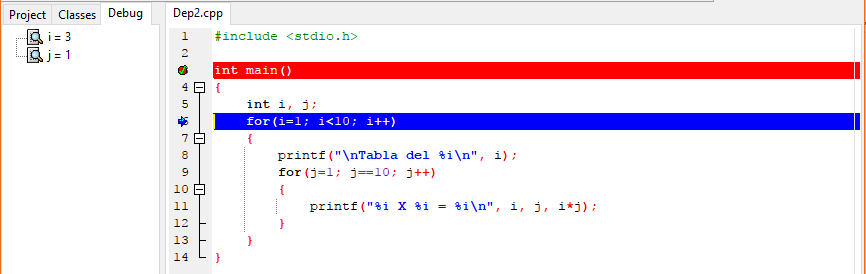


Al no entrar al ciclo while continua con el printf e imprime a la variable AS, la cual vale 1.



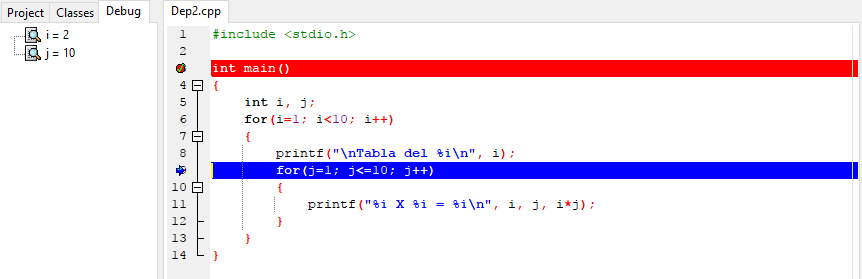
Posteriormente termina el programa.

**PROGRAMA 2: DEPURACION DE UN PROGRAMA QUE HACE LAS TABLAS DE MULTIPLICAR**



Este programa debería realizar las tablas de multiplicar del 1 al 10, sin embargo al ejecutar el código el programa solo realiza las condiciones del primer For y el segundo For jamas entra en funcionamiento, al analizar el programa nos damos cuenta que incrementa solo incrementa el primer For que es el que nos dice que de tabla de multiplicar se trata, sin embargo no ejecuta el segundo For que es el que se encarga de realizar la tabla de multiplicar del numero del primer For.



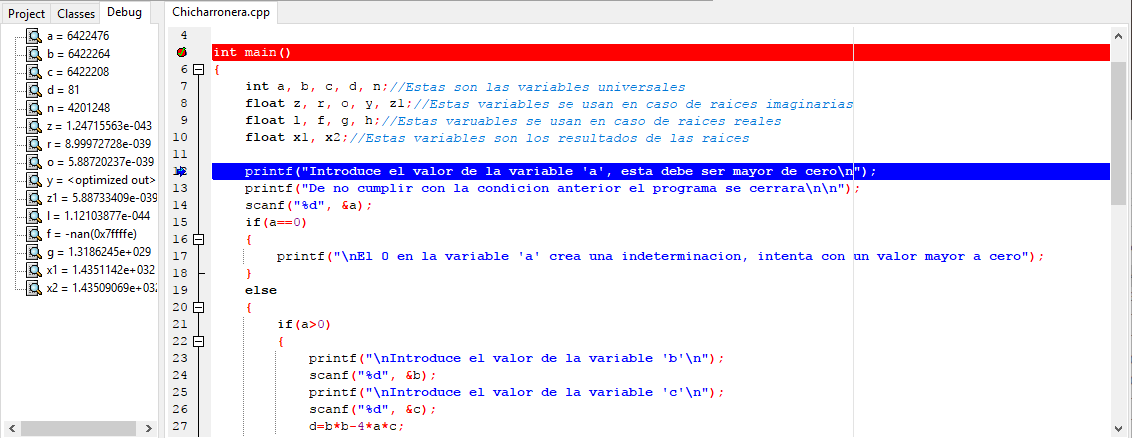


El problema se encuentra en que al momento de analizar la segunda condición del segundo For en el que nos dice que “j” debe ser igual a 10 contradice a la primer condición del mismo For que inicializa a “j” con 1, ya que en este punto la variable “j” es igual a 1, es imposible que en la segunda condición sea igual a 10, por lo tanto jamás avanza. Para corregir esto basta con asignar en la segunda condición que “j” debe de ser menor o igual a 10, pues de esta manera, permite que “j” acepte el valor de 1 anteriormente declarado y permita la ejecución normal del programa como se muestra a continuación.

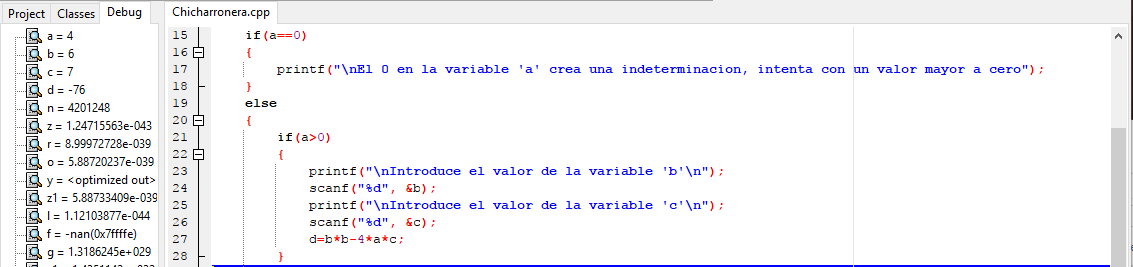


Al corregirse la condición en el segundo For permite el avance del código por lo que ahora en el ejecutable ya no solo nos va a mostrar de que tabla se trata, sino que también nos mostrara la tabla de multiplicar que antes no se nos mostraba.

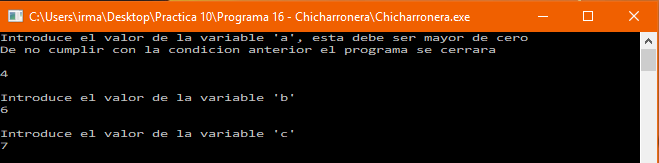
**PROGRAMA 3: DEPURACION DE PROGRAMA DE CHICHARRONERA.**

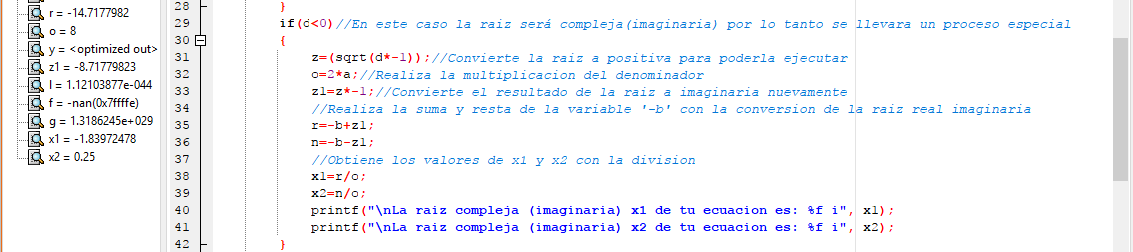


Al comenzar con la depuración el programa asigna valores al azar a las variables ya que estas aún no han tomado valores que el usuario introduzca o valores que se generen gracias a las operaciones que se realizaran en el programa, en este punto el ejecutable no tiene nada escrito.

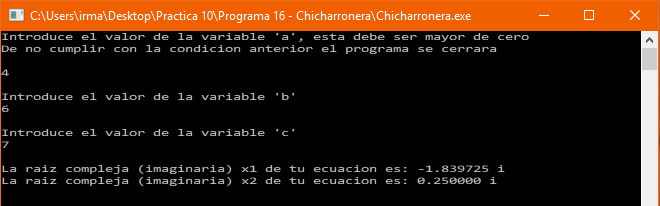


Se le pide al usuario ingresar un valor para la variable “a” el cual debe de ser diferente de cero, de lo contrario, el programa le mandara un aviso de que ingreso un 0 y terminara la ejecución, en este caso, ingresé como valor de “a” el número 4, por lo que continuó con el código pidiéndome ahora los valores de las variables “b” y “c” en los cuales ingrese 6 y 7 respectivamente.





Al tratarse de una combinación de números que como resultado me darían raíces complejas, el programa ejecuta el bloque de código que me permita cumplir con esta condición, por lo tanto al avanzar en el programa, cada una de las variables que me permitirán obtener valores de raíces complejas, iran adquiriendo valores específicos, de acuerdo a la o las operaciones que les corresponda ejecutar hasta llegar a los valores de x1 y x2.



Por lo tanto, al finalizar la depuración obtengo los resultados de x en números complejos o imaginarios.

CONCLUSION

A mi parecer el depurador es una excelente herramienta ya que gracias a ella podemos entender a la perfeccion el funcionamiento de cualquier programa y gracias a esto, permitirnos una facilidad de uso.

De la misma forma, la depuración nos ayuda a encontrar errores que el programa pueda tener y gracias a esto poder corregirlos para que el programa funcione con normalidad.