



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:

Ingeniera Claudia Rodriguez Espino

Asignatura:

Fundamentos de Programación

Grupo:

1102

No de Práctica(s):

Práctica número 10

Integrante(s):

Chaveste Bermejo Carlos Alberto

Semestre:

2018-1

Fecha de entrega:

27/10/2017

Observaciones:

CALIFICACIÓN: _____

OBJETIVO:

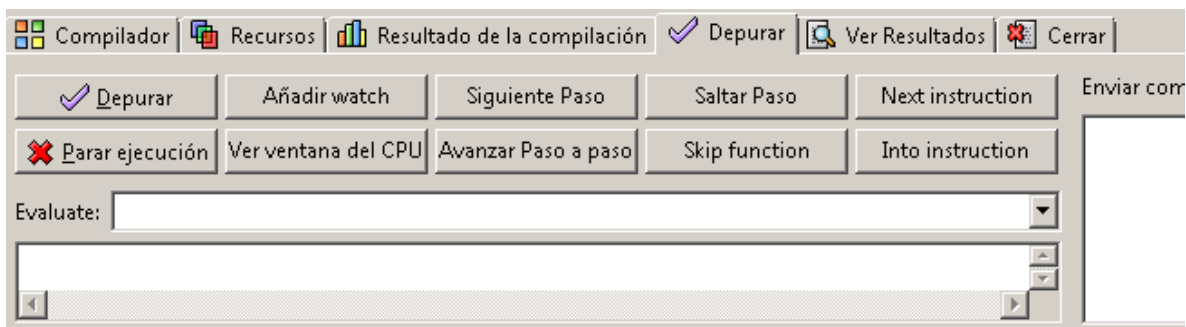
Aprender las técnicas básicas de depuración de programas en C para revisar de manera precisa el flujo de ejecución de un programa y el valor de las variables; en su caso, corregir posibles errores.

DESARROLLO:

En la práctica, aprendimos sobre los depuradores que sirven para someter a un programa aun ambiente de ejecución, que permite conocer el flujo del programa y conocer el comportamiento de las variables. Esta depuración es de vital importancia si se quiere:

- Optimizar el rendimiento del programa.
- Encontrar algún error de lógica o razonamiento.
- Detectar un error de ejecución o del compilador.

Cabe destacar, que se deben llevar a cabo una serie de pasos antes de depurar un programa adecuadamente, como lo son compilar y ejecutar. El depurador muestra datos importantes, como los puntos de ruptura (Que expresan a partir de donde en el código se va a depurar el programa), el valor de las variables y los pasos que se siguen en tiempo real. Se tienen opciones para visualizar adecuadamente el flujo del programa. En Dev C++ , se muestran opciones de: Depurar, Parar ejecución, Añadir watch, Ver Ventana del CPU, Siguiente Paso, Avanzar Paso a paso, Saltar paso, Saltar función, Siguiente instrucción y Dentro de la Instrucción, como se muestra en la interfaz de Dev C++.



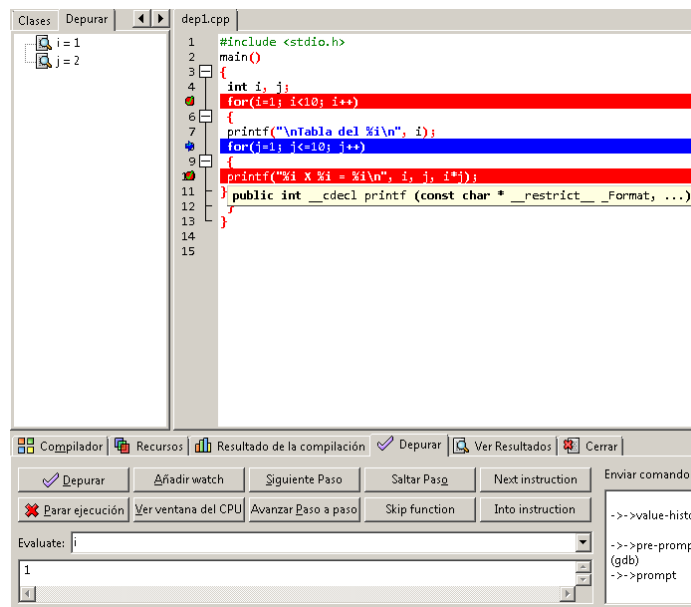
Para poder entrar al menú de depuración, es necesario que el compilador genere información de depuración, esto se hace entrando al menú: **Herramientas> Opciones del compilador> Configuración> Linker> Generar Información de Depuración> Aceptar.**

ACTIVIDAD EN CLASE

Programa 1, Tabla de cierto número.

```
#include <stdio.h>
main()
{
    int i, j;
    for(i=1; i<10; i++)
    {
        printf("\nTabla del %i\n", i);
        for(j=1; j<=10; j++)
        {
            printf("%i X %i = %i\n", i, j, i*j);
        }
    }
}
```

En el cuadro de depuración se puede agregar una variable de la siguiente forma

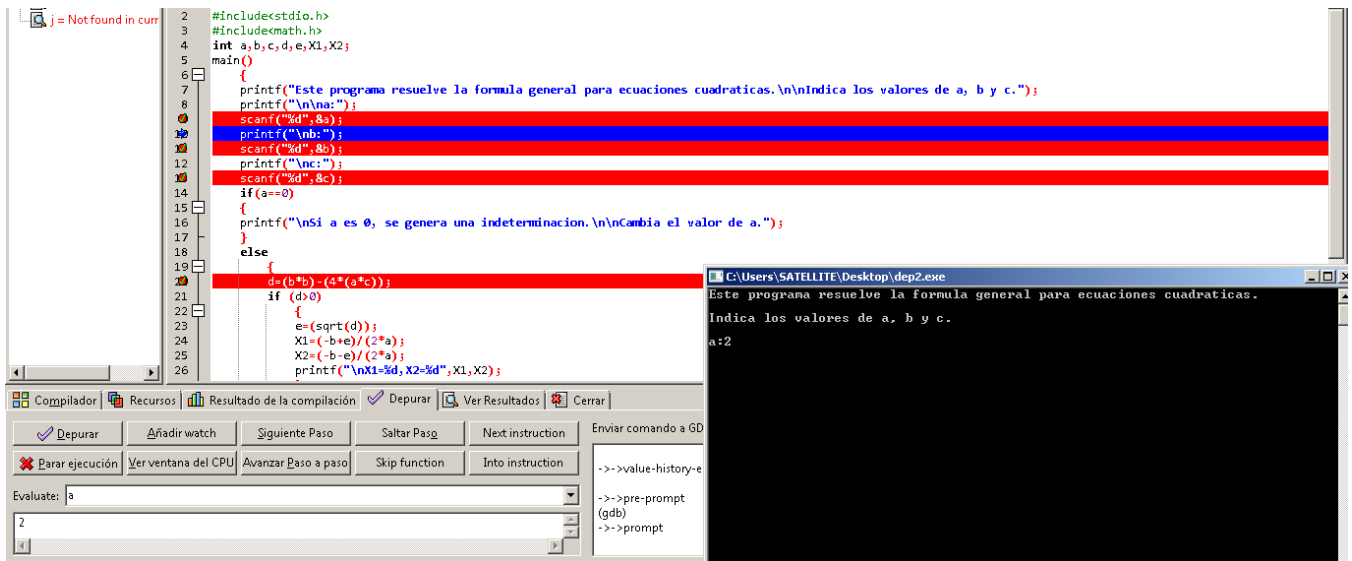


En la captura, se puede ver que el depurador muestra el valor de la variable i, en

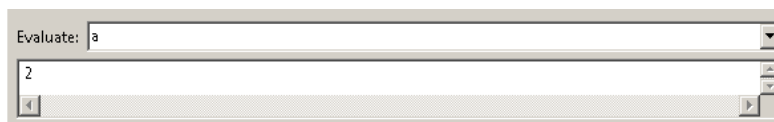
la quinta línea del código. En la ventana "Evaluate".

Programa 2. Fórmula general para resolver ecuaciones cuadráticas.

```
#include<stdio.h>
#include<math.h>
int a,b,c,d,e,X1,X2;
main()
{
    printf("Este programa resuelve la formula general para
ecuaciones cuadraticas.\n\nIndica los valores de a, b y c.");
    printf("\n\na:");
    scanf("%d",&a);
    printf("\nb:");
    scanf("%d",&b);
    printf("\nc:");
    scanf("%d",&c);
    if(a==0)
    {
        printf("\nSi a es 0, se genera una indeterminacion.\n\nCambia
el valor de a.");
    }
    else
    {
        d=(b*b)-(4*(a*c));
        if (d>0)
        {
            e=(sqrt(d));
            X1=(-b+e)/(2*a);
            X2=(-b-e)/(2*a);
            printf("\nX1=%d,X2=%d",X1,X2);
        }
        if(d<0)
        {
            d=d*-1;
            e=(sqrt(d));
            X1=(-b+e)/(2*a);
            X2=(-b-e)/(2*a);
            printf("\nX1=%d%di,X2=-%d%di",b,X1,b,X2);
        }
    }
}
```



En la captura se puede ver el valor que toma la variable a, y como el depurador puede visualizar esto a través de la variable que se agregó y la interfaz.



CONCLUSIÓN

El depurador puede ser una herramienta perfecta para que el flujo de un programa se vuelva más evidente, y de la misma forma, detectar errores de lógica o modificar el programa para obtener los resultados esperados.

BIBLIOGRAFÍA

<http://lcp02.fi-b.unam.mx/#>