



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

Marco Antonio Martinez Quintana

*Profesor:*

FUNDAMENTOS DE PROGRAMACION

*Asignatura:*

3

*Grupo:*

9

*No de Práctica(s):*

Andrik Uriel Reyes Roque

*Integrante(s):*

*No. de Equipo de  
cómputo empleado:*

No aplica

Fp03alu39

*No. de Lista o Brigada:*

1

*Semestre:*

11/12/2020

*Fecha de entrega:*

*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.

**Introducción**

Cuando un programa falla (no termina su ejecución de manera correcta), la mayoría de las veces, el compilador envía cierta información que permite determinar dónde se encuentra el problema. [...] no siempre es así y es más complicado saber dónde falla el programa y cómo corregirlo. En esos casos es mejor ejecutar el programa en un contexto controlado para saber, exactamente, donde está fallando. La depuración se refiere al hecho de poder ejecutar un programa en un ambiente controlado de manera que se pueda estudiar el flujo de ejecución

**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.

**Ejercicios propuestos**

Para el siguiente código fuente, utilizar algún entorno de depuración para encontrar la utilidad del programa y la funcionalidad de los principales comandos de depuración, como puntos de ruptura, ejecución de siguiente línea o instrucción.

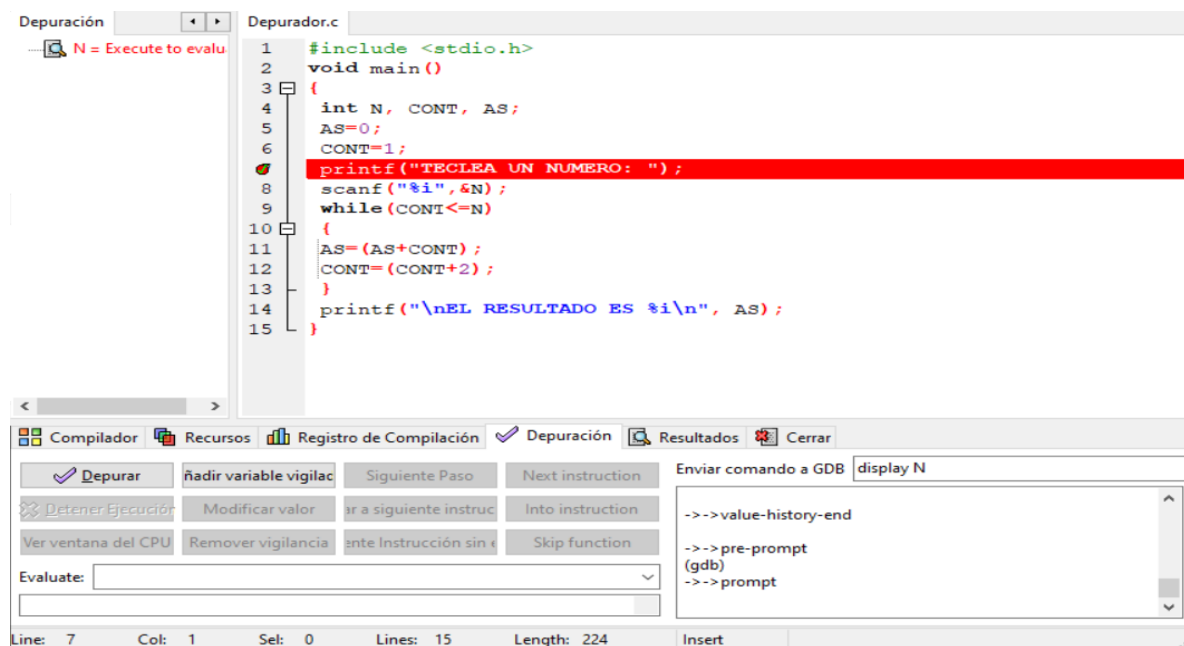
```

#include <stdio.h>

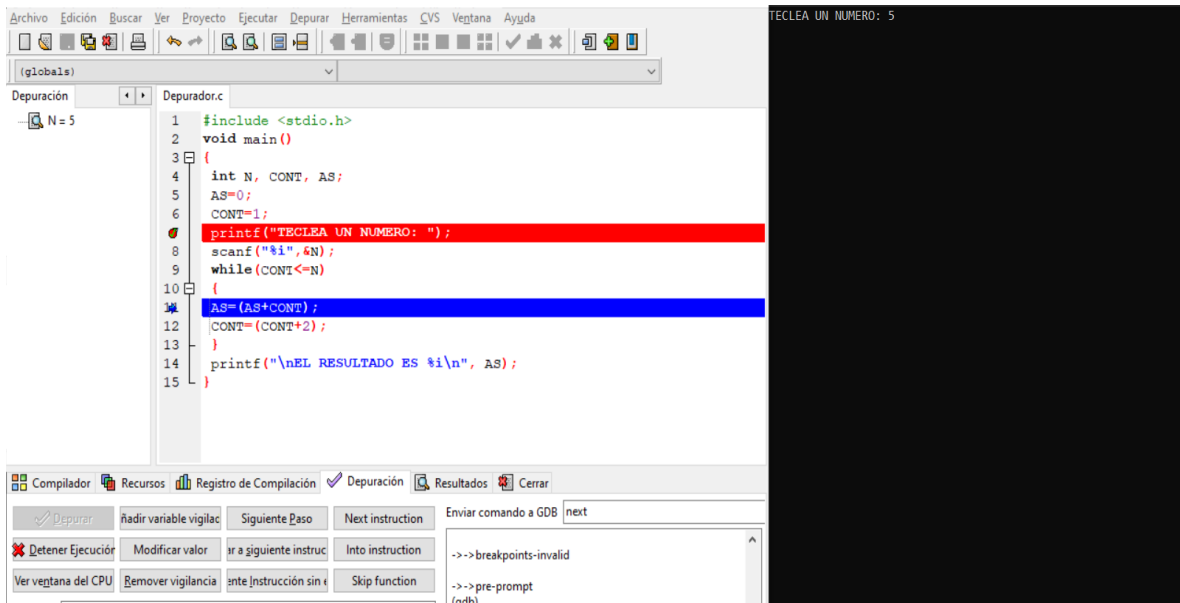
void main()
{
    int N, CONT, AS;
    AS=0;
    CONT=1;
    printf("TECLEA UN NUMERO: ");
    scanf("%i",&N);
    while(CONT<=N)
    {
        AS=(AS+CONT);
        CONT=(CONT+2);
    }
    printf("\nEL RESULTADO ES %i\n", AS);
}

```

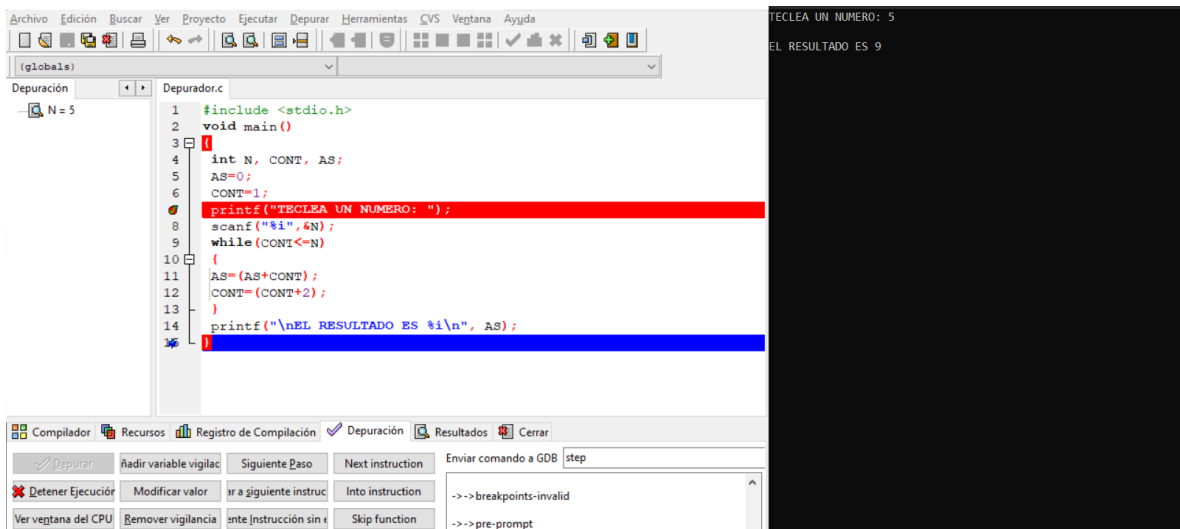
Antes de depurar el programa, podemos seleccionar un apartado conocido como punto de ruptura en donde se empezará a ejecutar al momento de depurar. También se puede seleccionar una variable e ir viendo sus cambios.



Cuando se depura, inicia en el renglón seleccionado y conforme le das en siguiente paso vas añadiendo en la consola lo requerido y como se ve en la imagen, la variable cambia al dígito asignado



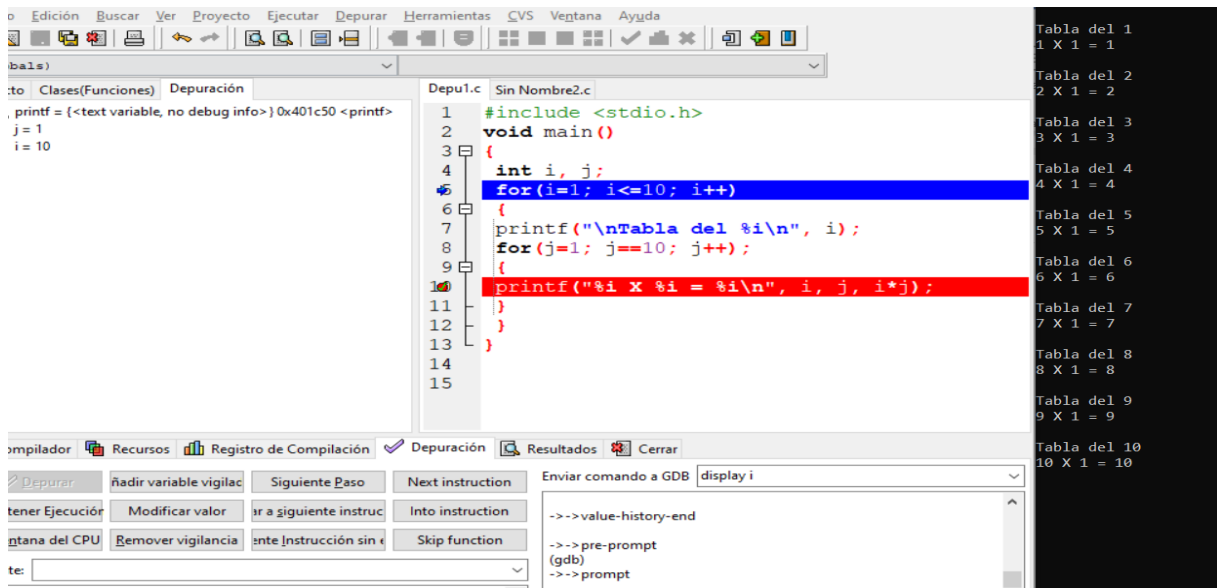
Cada vez que usas la opción “siguiente paso” te va mostrando todos los procesos que ejecuta hasta llegar al resultado



Funcionalidad del programa:

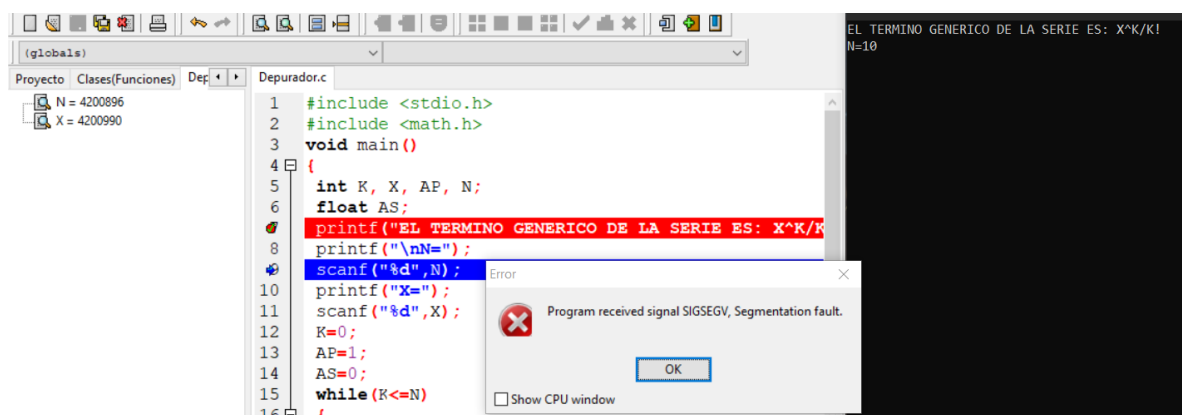
El programa se encargar de Agrupar pares de números ((1,2),(3,4),(5,6)(...)) en donde empezando por el 1 y va aumentando de 2 en 2 el resultado

**-El siguiente programa debe mostrar las tablas de multiplicar desde la del 1 hasta la del 10.**

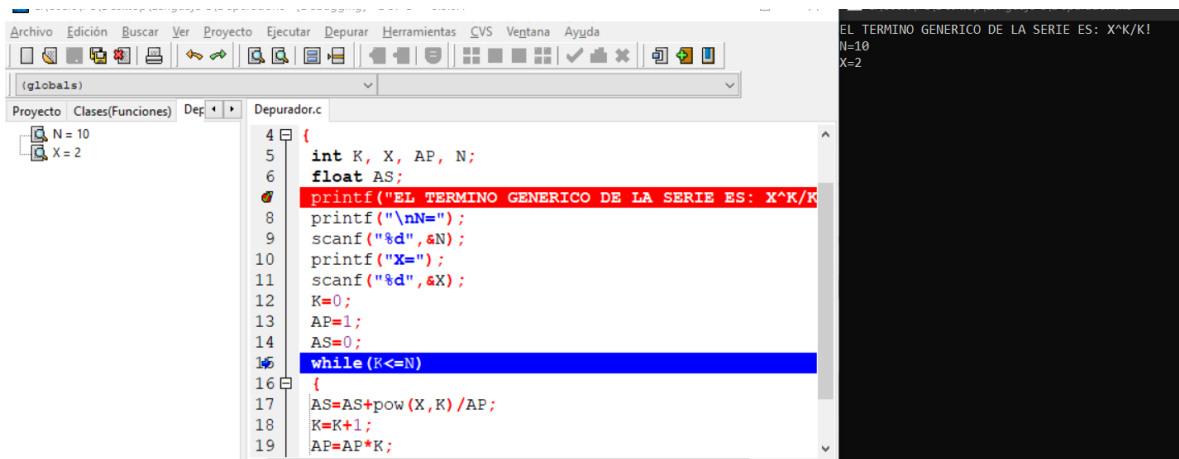


El programa no mostraba las tablas debido a que no había un “;” terminando la columna del segundo for y solamente repetía el primer for con el printf. La variable i ponía que debía ser menor que 10, por esa razón las tablas solo llegaban hasta el 9.

**-El siguiente programa muestra una violación de segmento durante su ejecución y se interrumpe; usar un depurador para detectar y corregir la falla.**



El problema esta en que al momento de introducir el numero este no lo guardaba y por eso aparecía el error



```
4 {  
5     int K, X, AP, N;  
6     float AS;  
7     printf("EL TERMINO GENERICO DE LA SERIE ES: X^K/K!");  
8     printf("\nN=");  
9     scanf("%d", &N);  
10    printf("X=");  
11    scanf("%d", &X);  
12    K=0;  
13    AP=1;  
14    AS=0;  
15    while (K<=N)  
16    {  
17        AS=AS+pow(X,K)/AP;  
18        K=K+1;  
19        AP=AP*K;
```

EL TERMINO GENERICO DE LA SERIE ES: X^K/K!  
N=10  
X=2

Se tiene que utilizar el **&** para guardar las variables y así poder seguir ejecutando el programa

## Conclusión:

Ahora que conozco cual es la función de estos programas para depurar, veo que son demasiado importantes ya que te permiten estar checándolo línea por línea cualquier error que se pueda encontrar durante la ejecución del programa y este te ayuda ya que te va mostrando paso a paso como es que se desarrolla y con ayuda de las herramientas que tiene , puedes ir viendo que valores toma cada variable y encontrar errores específicos en lugares donde ya tengas un punto marcado y visualizar que pasos se llega a saltar o si falta algún carácter a poner. Este programa lo usare para la creación de mi proyecto final y a futuro ya que será como una guía donde podrás ver todos los errores y resolverlos,

## Bibliografía

E., E., & A., J. (s. f.). Guía práctica de estudio: Depuración de programas. UNAM. Recuperado 13 de diciembre de 2020, de <http://odin.fi-b.unam.mx/salac/practicafp/Anexo%20DepuracionProgramas.pdf>