



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Marco Antonio Martínez Quintana

*Asignatura:* Fundamentos de Programación

*Grupo:* 3

*No de Práctica(s):* 10

*Integrante(s):* Ramírez Pérez Daniela Itzel

*No. de Equipo de  
cómputo empleado:* No aplica

*No. de Lista o Brigada:* 36

*Semestre:* 1er

*Fecha de entrega:* Viernes 4 de Diciembre

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## **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.

### **Introducción**

Depurar un programa significa someterlo a un ambiente de ejecución controlado para conocer cada aspecto del flujo de ejecución del programa.

#### **Razones para usar un depurador:**

- Se desea optimizar el programa
- El programa tiene algún fallo
- El programa tiene un error de ejecución o defecto

#### **Funciones básicas**

- Ejecutar y mostrar el código fuente del programa para mostrar que parte se esta ejecutando
- Punto de ruptura es un indicador para detener la ejecución del programa en algún punto indicado
- Continuar con la ejecución del programa después del punto de ruptura.
- Ejecutar la siguiente instrucción: sirve para ejecutar una instrucción más y detener el programa de nuevo
- Ejecutar la siguiente línea: es muy similar a la función anterior, pero realizará todas las instrucciones necesarias hasta llegar a la siguiente línea de código. S
- Ejecutar la instrucción o línea anterior:
- Visualizar el valor de las variables

Dependiendo de la herramienta usada para compilar el programa, si es de consola o de terminal, su uso y las funciones disponibles variarán.

## Ejercicios propuestos

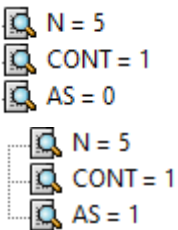
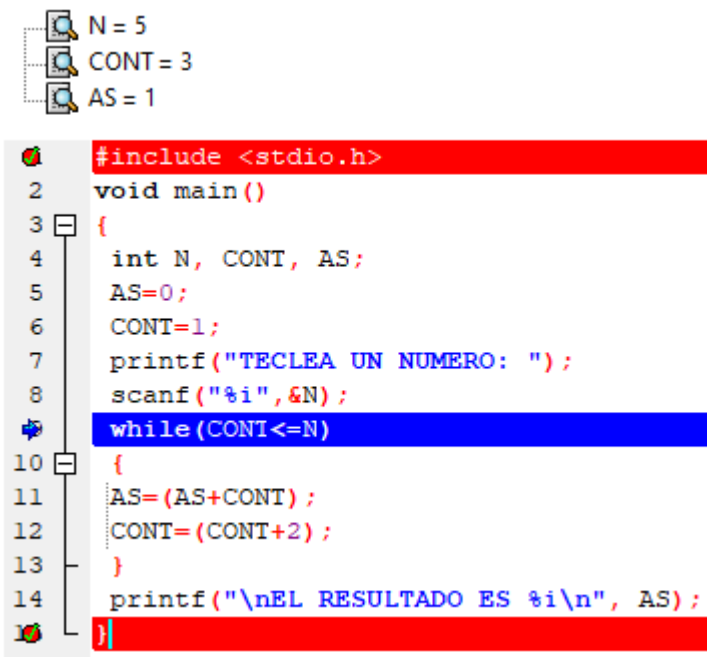
Para el siguiente código fuente, utilizar algún entorno de depuración para encontrar la utilidad del programa y la funcionalidad de los principales comandos de depuración, como puntos de ruptura, ejecución de siguiente línea o instrucción.

```
#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);
}
```

```
1  #include <stdio.h>
2  void main()
3  {
4      int N, CONT, AS;
5      AS=0;
6      CONT=1;
7      printf("TECLEA UN NUMERO: ");
8      scanf("%i",&N);
9      while(CONT<=N)
10     {
11         AS=(AS+CONT);
12         CONT=(CONT+2);
13     }
14     printf("\nEL RESULTADO ES %i\n", AS);
15 }
```

TECLEA UN NUMERO: 5

EL RESULTADO ES 9

Paso	Explicacion
 <pre> 1 N = 5 2 CONT = 1 3 AS = 0 4 5 N = 5 6 CONT = 1 7 AS = 1 </pre>	<p>Para empezar las variables son establecidas.</p> <p>Con la función while se le suma 0+1 resultando en la variable AS=1</p>
 <pre> 1 #include &lt;stdio.h&gt; 2 void main() 3 { 4     int N, CONT, AS; 5     AS=0; 6     CONT=1; 7     printf("TECLEA UN NUMERO: "); 8     scanf("%i",&amp;N); 9     while (CONT&lt;=N) 10    { 11        AS=(AS+CONT); 12        CONT=(CONT+2); 13    } 14    printf("\nEL RESULTADO ES %i\n", AS); 15 } </pre>	<p>El siguiente paso es con la función while checando si es verdadera la expresión que dice que CONT debe ser menor o igual a N en este caso siendo verdadero significando que se realizaran las acciones dentro del bloque sumándosele a CONT 2 resultando al cambio de 1 a 3.</p>

N = 5  
CONT = 3  
AS = 4

```
#include <stdio.h>
2 void main()
3 {
4     int N, CONT, AS;
5     AS=0;
6     CONT=1;
7     printf("TECLEA UN NUMERO: ");
8     scanf("%i",&N);
9     while (CONT<=N)
10    {
11        AS=(AS+CONT);
12        CONT=(CONT+2);
13    }
14    printf("\nEL RESULTADO ES %i\n", AS);
15 }
```

La siguiente instrucción es que a AS se le debe de Sumar lo que obtuvimos en CONT resultando en el cambio de variable AS=1 a AS=4 y sumándosele nuevamente s CONT+2

N = 5  
CONT = 5  
AS = 4

```
#include <stdio.h>
2 void main()
3 {
4     int N, CONT, AS;
5     AS=0;
6     CONT=1;
7     printf("TECLEA UN NUMERO: ");
8     scanf("%i",&N);
9     while (CONT<=N)
10    {
11        AS=(AS+CONT);
12        CONT=(CONT+2);
13    }
14    printf("\nEL RESULTADO ES %i\n", AS);
15 }
```

El bloque de instrucciones continua y a CONT se le suma de nuevo 2 y se le vuelve a comparar con la expresión lógica de while y sigue siendo verdadera por lo que se repite el bloque.

N = 5  
CONT = 5  
AS = 9

```
#include <stdio.h>
2 void main()
3 {
4     int N, CONT, AS;
5     AS=0;
6     CONT=1;
7     printf("TECLEA UN NUMERO: ");
8     scanf("%i",&N);
9     while (CONT<=N)
10    {
11        AS=(AS+CONT);
12        CONT=(CONT+2);
13    }
14    printf("\nEL RESULTADO ES %i\n", AS);
}
```

A la variable AS se le suma lo que contiene la variable CONT resultando en AS=9

N = 5  
CONT = 7  
AS = 9

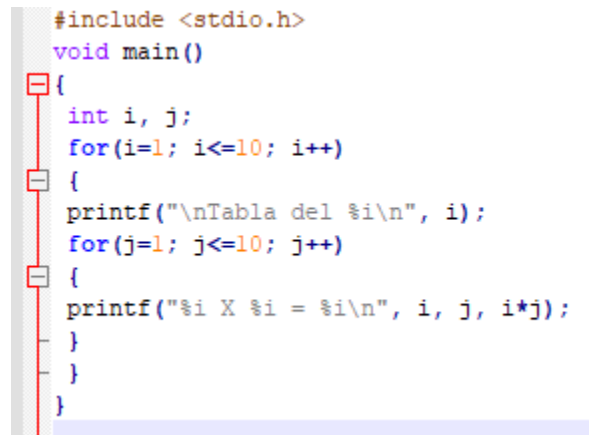
```
#include <stdio.h>
2 void main()
3 {
4     int N, CONT, AS;
5     AS=0;
6     CONT=1;
7     printf("TECLEA UN NUMERO: ");
8     scanf("%i",&N);
9     while (CONT<=N)
10    {
11        AS=(AS+CONT);
12        CONT=(CONT+2);
13    }
14    printf("\nEL RESULTADO ES %i\n", AS);
}
```

Y sigue la siguiente instrucción sumándosele a CONT 2 que resulta en CONT=7 y se vuelve a checar en while si es verdadero en esta caso ya la expresión se le considera falso y se procede con el resto del programa resultando en la impresión del número obtenido siendo 9 .

El siguiente programa debe mostrar las tablas de multiplicar desde la del 1 hasta la del 10. En un principio no se mostraba la tabla del 10, luego después de intentar corregirse sin un depurador dejaron de mostrarse el resto de las tablas. Usar un depurador de C para averiguar el funcionamiento del programa y corregir ambos problemas.

```
#include <stdio.h>
void 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);
        }
    }
}
```

### Corrección



```
#include <stdio.h>
void 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);
        }
    }
}
```

Original	Corrección	Explicación
<code>for(i=1; i&lt;10; i++)</code>	<code>for(i=1; i&lt;=10; i++)</code>	En esta expresión el error que se cometió es que <code>i&lt;10</code> lo que causaba que solo se imprimieran hasta la tabla del 9
<code>for(j=1; j==10; j++)</code>	<code>for(j=1; j&lt;=10; j++)</code>	En la expresión original el error cometido fue <code>j==10</code> que causaba un error al ser establecido que <code>j</code> era 1 establecía que la expresión era falsa al no ser igual a 10.

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

```
#include <stdio.h>
#include <math.h>
void main()
{
    int K, X, AP, N;
    float AS;
    printf("EL TERMINO GENERICO DE LA SERIE ES: X^K/K!");
    printf("\nN=");
    scanf("%d",N);
    printf("X=");
    scanf("%d",X);
    K=0;
    AP=1;
    AS=0;
    while(K<=N)
    {
        AS=AS+pow(X,K)/AP;
        K=K+1;
        AP=AP*K;
    }
    printf("SUM=%le",AS);
}
```

El problema en el código surge que al querer registrar lo que escribe el usuario, debido a que se omitió el signo “&” desencadenando una serie de errores en del programa.

## Corrección

```
#include <stdio.h>
#include <math.h>
void main()
{
    int K, X, AP, N;
    float AS;
    printf("EL TERMINO GENERICO DE LA SERIE ES: X^K/K!\n");
    printf("N=");
    scanf("%d",&N);
    printf("X=");
    scanf("%d",&X);
    K=0;
    AP=1;
    AS=0;
    while(K<=N)
    {
        AS=AS+pow(X,K)/AP;
        K=K+1;
        AP=AP*K;
    }
    printf("SUM=%le",AS);
}
```

```
EL TERMINO GENERICO DE LA SERIE ES: X^K/K!
N=2
X=3
SUM=8.500000e+000
```



## **Conclusión**

Gracias a la practica se pudo reforzar los conocimientos adquiridos para visualizar errores que se encontraban en los ejercicios y ya se conocen las herramientas básicas de los depuradores que nos será útil cuando se estén teniendo problemas con nuestros demás programas.

## **Referencias**

- Nakayama Cervantes, A., Castañeda Perdomo, M., Solano Gálvez, J. A., García Cano, E. E., Sandoval Montaña, L., & Arteaga Ricci, T. I. (2018, 6 abril). Manual de prácticas del laboratorio de Fundamentos de programación (p.135-147). Laboratorio de Computación Salas A y B. [http://odin.fib.unam.mx/salac/practicasFP/MADO-17\\_FP.pdf](http://odin.fib.unam.mx/salac/practicasFP/MADO-17_FP.pdf)