	<b>Carátula para entrega de prácticas</b>	
Facultad de Ingeniería	Laboratorio de docencia	

# Laboratorios de computación salas A y B

<i>Profesor:</i>	Marco Antonio Martínez Quintana
<i>Asignatura:</i>	Fundamentos de Programación
<i>Grupo:</i>	3
<i>No de Práctica(s):</i>	Práctica 10
<i>Integrante(s):</i>	Camacho Bernabé Roberto Ángel
<i>No. de Equipo de cómputo empleado:</i>	No aplica
<i>No. de Lista o Brigada:</i>	3
<i>Semestre:</i>	2021-1
<i>Fecha de entrega:</i>	14 de diciembre de 2020
<i>Observaciones:</i>	

**CALIFICACIÓN:** \_\_\_\_\_

## Guía práctica de estudio 10: 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 por medio de herramientas dedicadas a ello. Este ambiente permite conocer exactamente el flujo de ejecución del programa, el valor que las variables adquieren, la pila de llamadas a funciones, entre otros aspectos. Es importante poder compilar el programa sin errores antes de depurarlo.

Antes de continuar, es necesario conocer las siguientes definiciones (extraídas del Glosario IEEE610) ya que son parte latente del proceso de Desarrollo de Software:

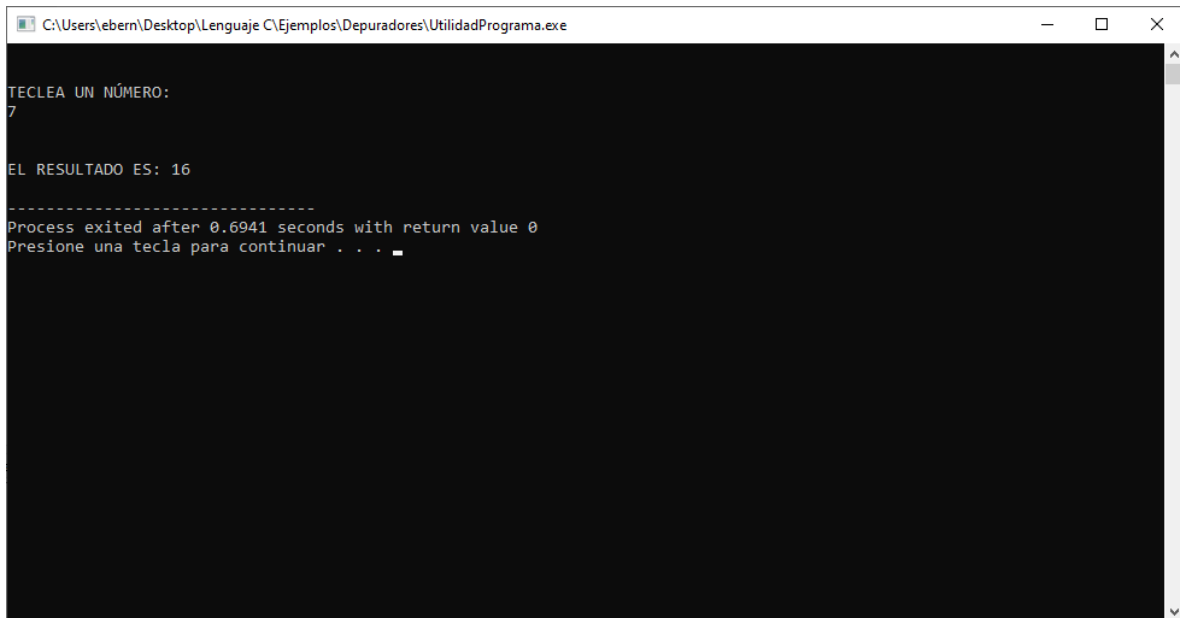
Error. Se refiere a una acción humana que produce o genera un resultado incorrecto. Defecto (Fault). Es la manifestación de un error en el software. Un defecto es encontrado porque causa una Falla (failure). Falla (failure). Es una desviación del servicio o resultado esperado.

La depuración de un programa es útil cuando:

- ♣ Se desea optimizar el programa: no basta que el programa se pueda compilar y se someta a pruebas que demuestren que funciona correctamente. Debe realizarse un análisis exhaustivo del mismo en ejecución para averiguar cuál es su flujo de operación y encontrar formas de mejorarlo (reducir el código, utilizar menos recursos llegando a los mismos resultados, hacer menos rebuscado al algoritmo), o bien, encontrar puntos donde puede fallar con ciertos tipos de entrada de datos.
- ♣ El programa tiene algún fallo: el programa no muestra los resultados que se esperan para cierta entrada de datos debido a que el programador cometió algún error durante el proceso de diseño. Muchas veces encontrar este tipo de fallos suele ser difícil, ya sea porque la percepción del programador no permite encontrar la falla en su diseño o porque la errata es muy pequeña, pero crucial. En este caso es de mucha utilidad conocer paso a paso cómo se ejecutan las estructuras de control, qué valor adquieren las variables, etc.
- ♣ El programa tiene un error de ejecución o defecto: cuando el programa está ejecutándose, éste se detiene inesperadamente. Suele ocurrir por error en el diseño o implementación del programa en las que no se contemplan las limitaciones del lenguaje de programación o el equipo donde el programa se ejecuta. Como el programa se detiene inesperadamente, no se conoce la parte del programa donde

se provoca el defecto, teniendo que recurrir a la depuración para encontrarlo. El más común de este tipo de defecto es la “violación de segmento”.

Ejercicios propuestos en la práctica:



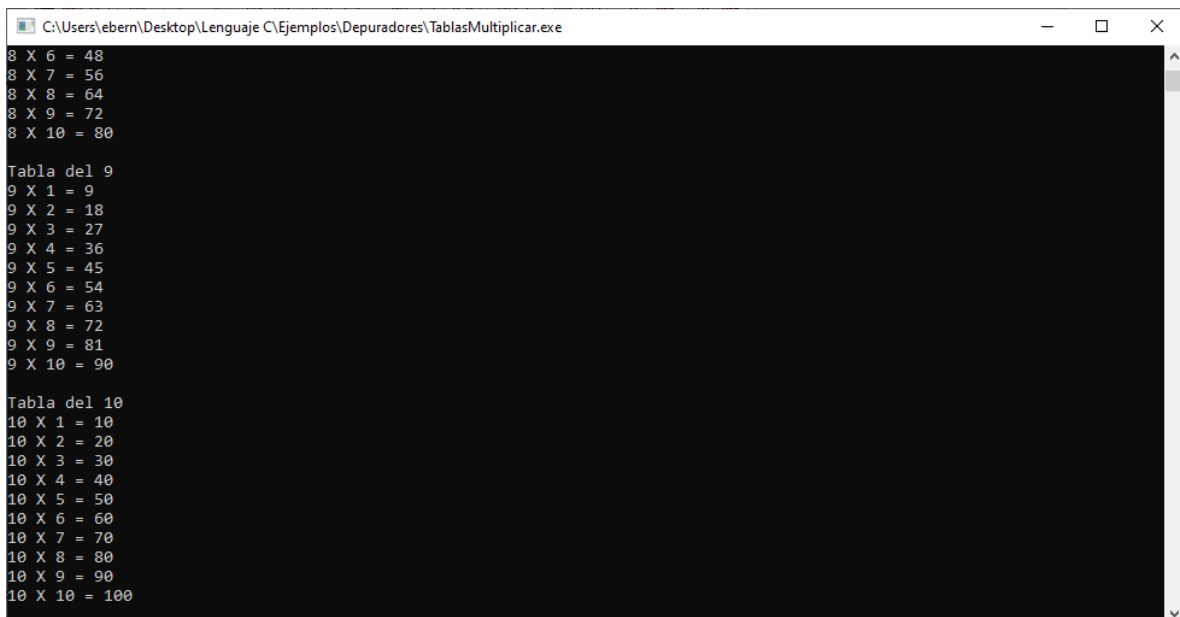
```
C:\Users\ebem\Desktop\Lenguaje C\Ejemplos\Depuradores\UtilidadPrograma.exe

TECLEA UN NÚMERO:
7

EL RESULTADO ES: 16

-----
Process exited after 0.6941 seconds with return value 0
Presione una tecla para continuar . . .
```

Imagen 1. Uso del ciclo de repetición While para la suma de dos números.



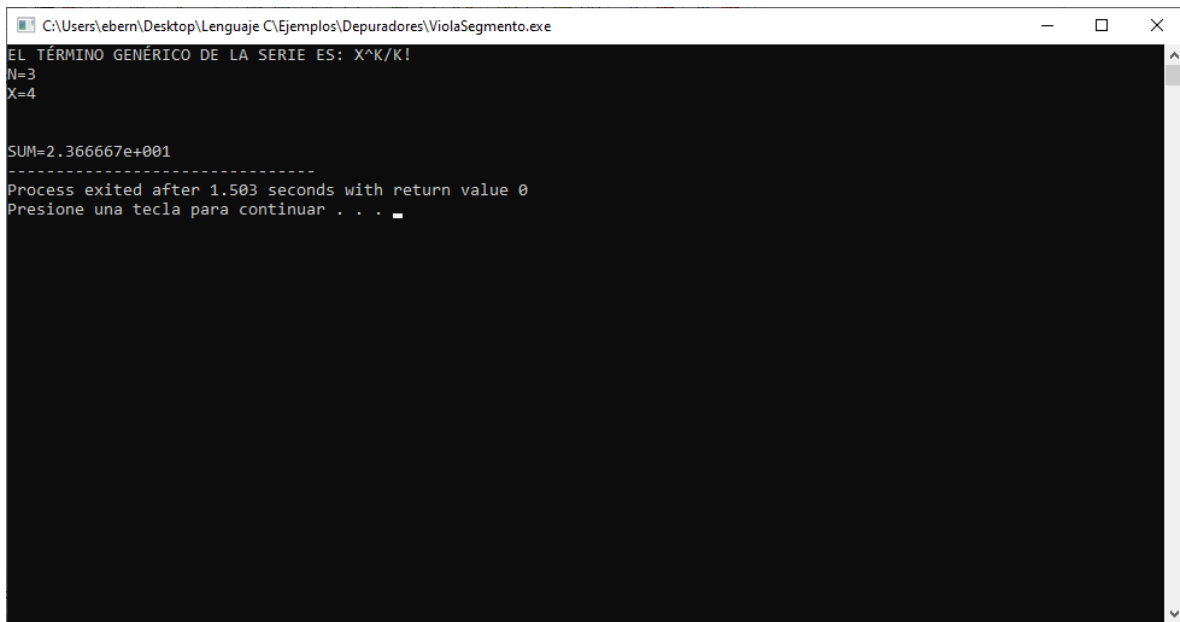
```
C:\Users\ebem\Desktop\Lenguaje C\Ejemplos\Depuradores\TablasMultiplicar.exe

8 X 6 = 48
8 X 7 = 56
8 X 8 = 64
8 X 9 = 72
8 X 10 = 80

Tabla del 9
9 X 1 = 9
9 X 2 = 18
9 X 3 = 27
9 X 4 = 36
9 X 5 = 45
9 X 6 = 54
9 X 7 = 63
9 X 8 = 72
9 X 9 = 81
9 X 10 = 90

Tabla del 10
10 X 1 = 10
10 X 2 = 20
10 X 3 = 30
10 X 4 = 40
10 X 5 = 50
10 X 6 = 60
10 X 7 = 70
10 X 8 = 80
10 X 9 = 90
10 X 10 = 100
```

Imagen 2. Tabla de multiplicar del 1 al 10 con el error arreglado.



```
C:\Users\eborn\Desktop\Lenguaje C\Ejemplos\Depuradores\ViolaSegmento.exe
EL TÉRMINO GENÉRICO DE LA SERIE ES: X^K/K!
N=3
X=4

SUM=2.366667e+001
-----
Process exited after 1.503 seconds with return value 0
Presione una tecla para continuar . . . _
```

Imagen 3. Uso del ciclo de repetición While para elevar cierto número.

## Conclusiones

Los depuradores permiten y facilitan al programador la acción de programar cualquier código ya que con las diferentes funciones que tienen permite la visualización y un mayor entendimiento del código. En caso de que exista un error, necesites hacer puntos de ruptura o algo entre en conflicto entre las funciones que estamos trabajando dichos depuradores permitirán el arreglo inmediato de dicho problema.

## Referencias

Facultad de Ingeniería. (2018, 6 abril). Recuperado 11 de diciembre de 2020. Guía práctica de estudio 10: Depuración de programas. Área/Departamento: Laboratorio de computación salas A y B.