Universidad Nacional Autónoma de México

Facultad de Ingeniería

Fundamentos de programación

Semestre 2020-1

Práctica 10

Depuración de programas

Matias Zavala Melissa Maruuati

M.I. Marco Antonio Martínez Quintana

30-diciembre-2020

**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**

**Depuración de programas escritos en C con GCC y GDB**

Para depurar un programa usando las herramientas desarrolladas por GNU, éste debe compilarse con información para depuración por medio del compilador GCC.

Para compilar, “nombre”.c con GCC con información de depuración, en una terminal con el siguiente comando:

gcc -g -o “nombre” “nombre”.c

El parámetro -g indica que se ejecuta como depuración.

Para depurar con GDB

gdb ./calculadora

Corriendo GDB se entra a una línea de comandos:

list o l: Permite listar diez líneas del código fuente del programa o un rango de estas.

b: Establece un punto de ruptura para lo cual debe indicarse en qué línea se desea establecer o el nombre de la función

d o delete: Elimina un punto de ruptura, indicando cuál es el que debe eliminarse usando el número de línea.

clear: Elimina todos los puntos de ruptura.

info line: Muestra la información acorde a la línea indicada después del comando.

run o r: Ejecuta el programa. Si el programa tiene un punto de ruptura se ejecutará hasta ahi, en caso contrario se ejecutará todo el programa.

c: Continúa con la ejecución del programa después de un punto de ruptura.

s: Continúa con la siguiente instrucción después de un punto de ruptura.

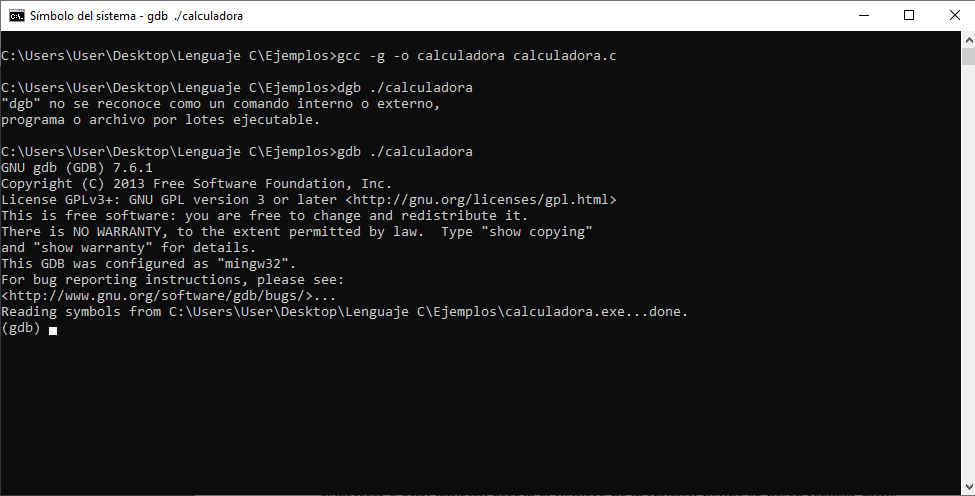
n: Salta hasta la siguiente línea de código después de un punto de ruptura.

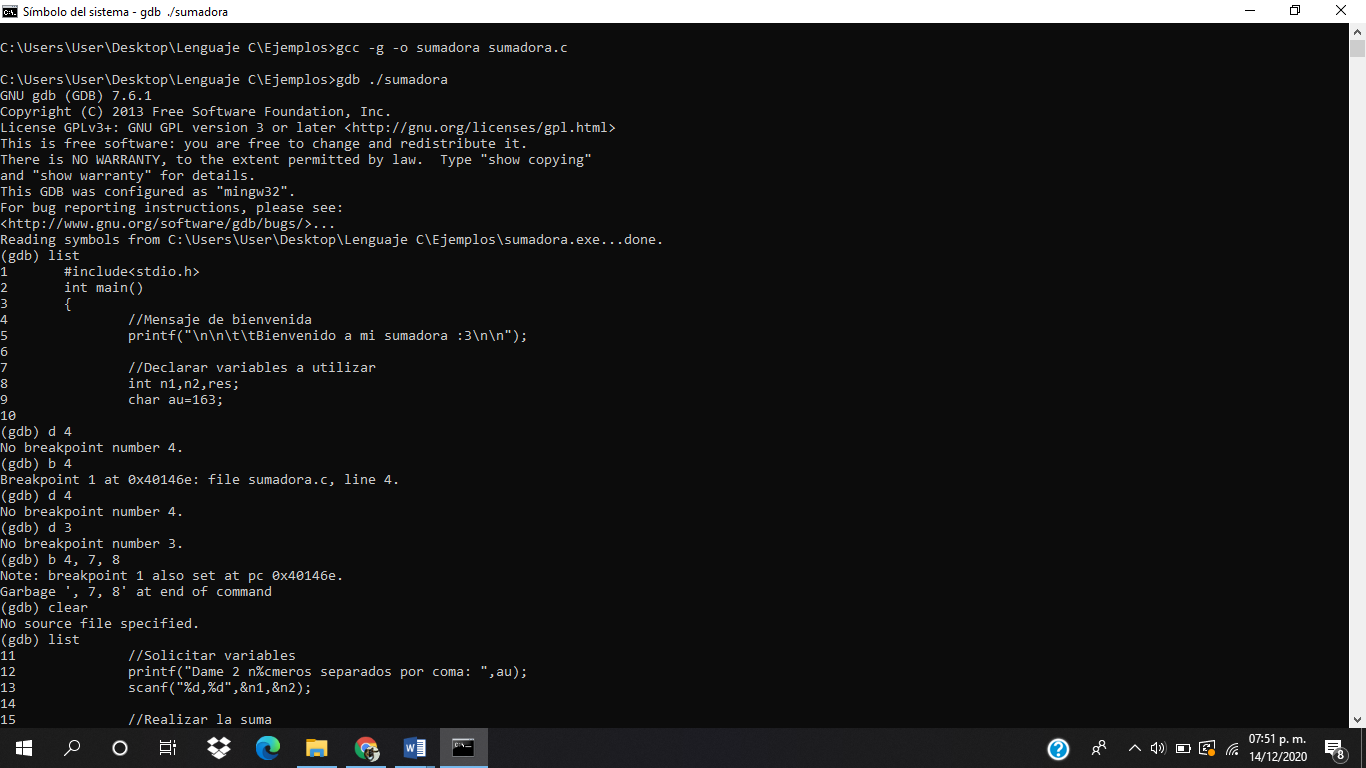
p o print: Muestra el valor de una variable.

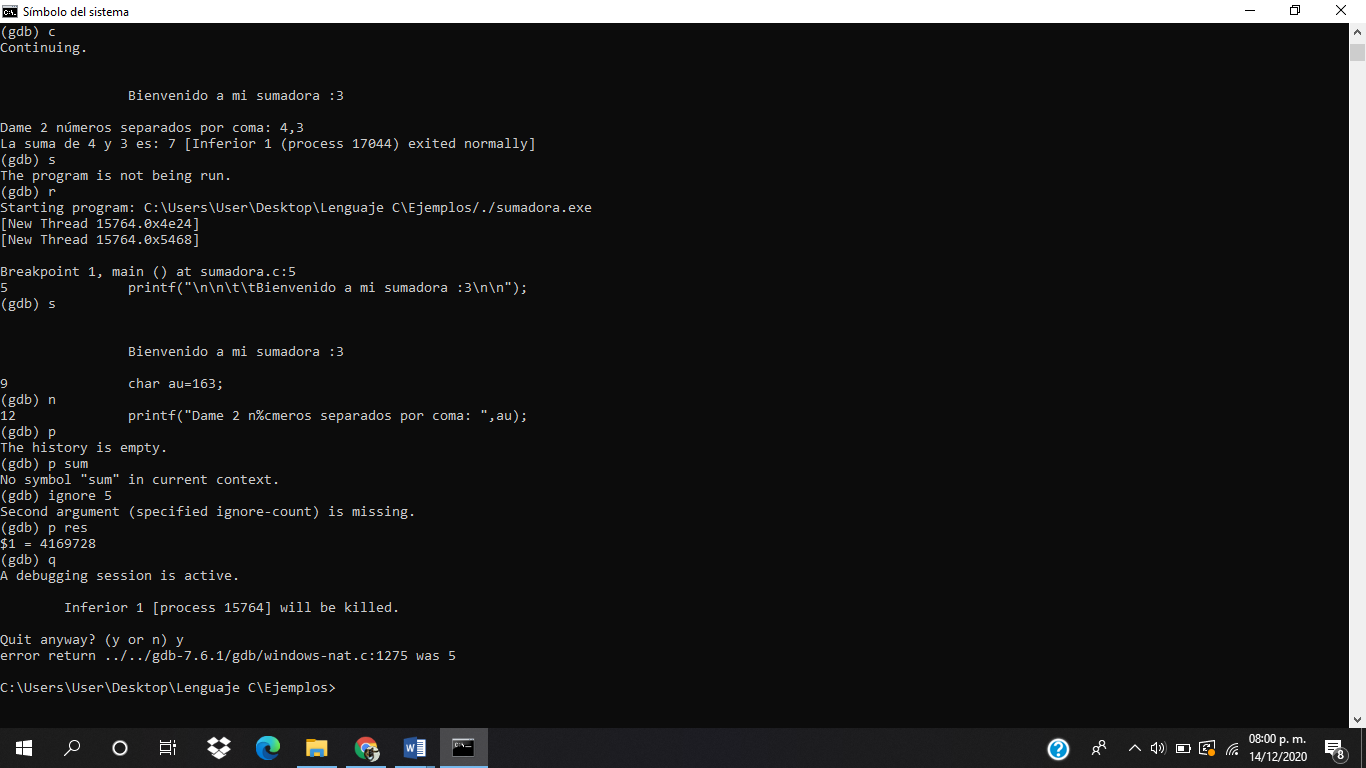
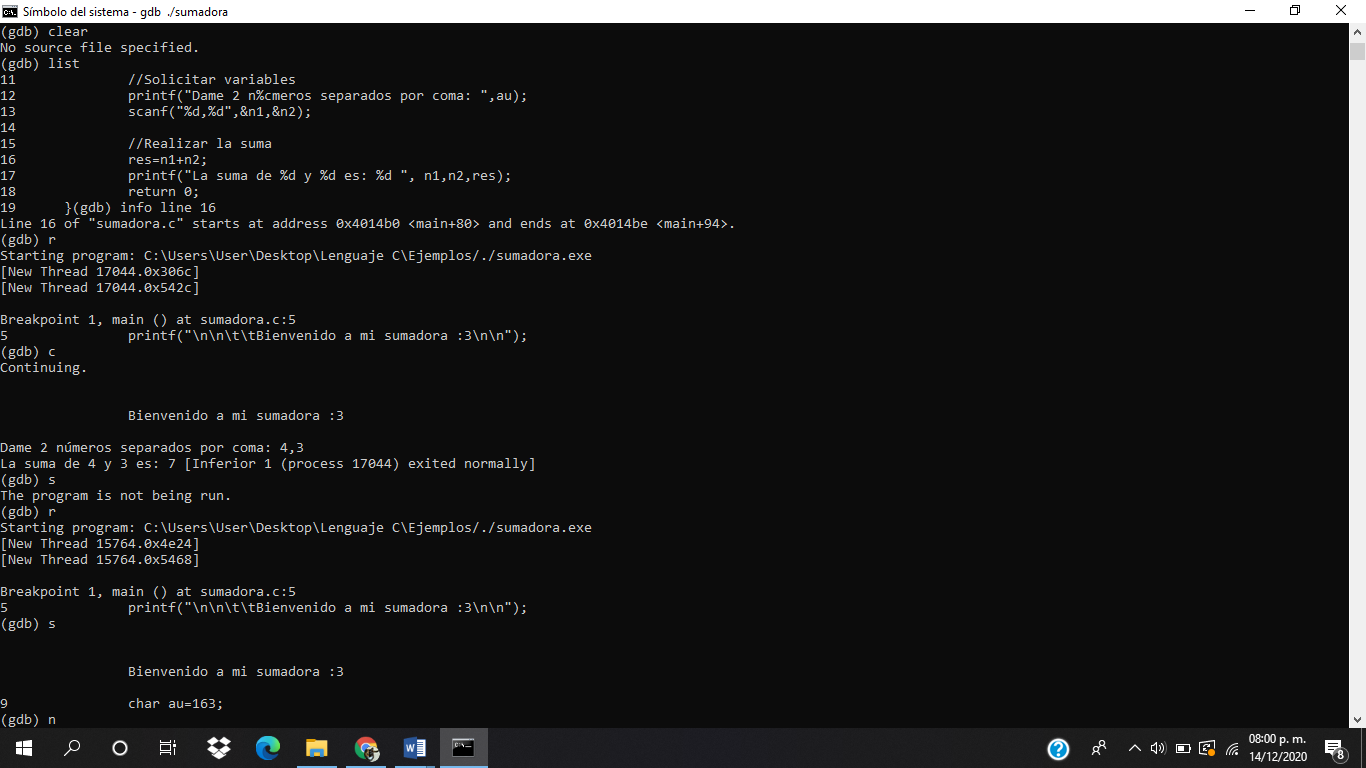
ignore: Ignora un determinado punto de ruptura.

q o quit: Termina la ejecución de GDB.

**Depuración del programa de sumadora.c**

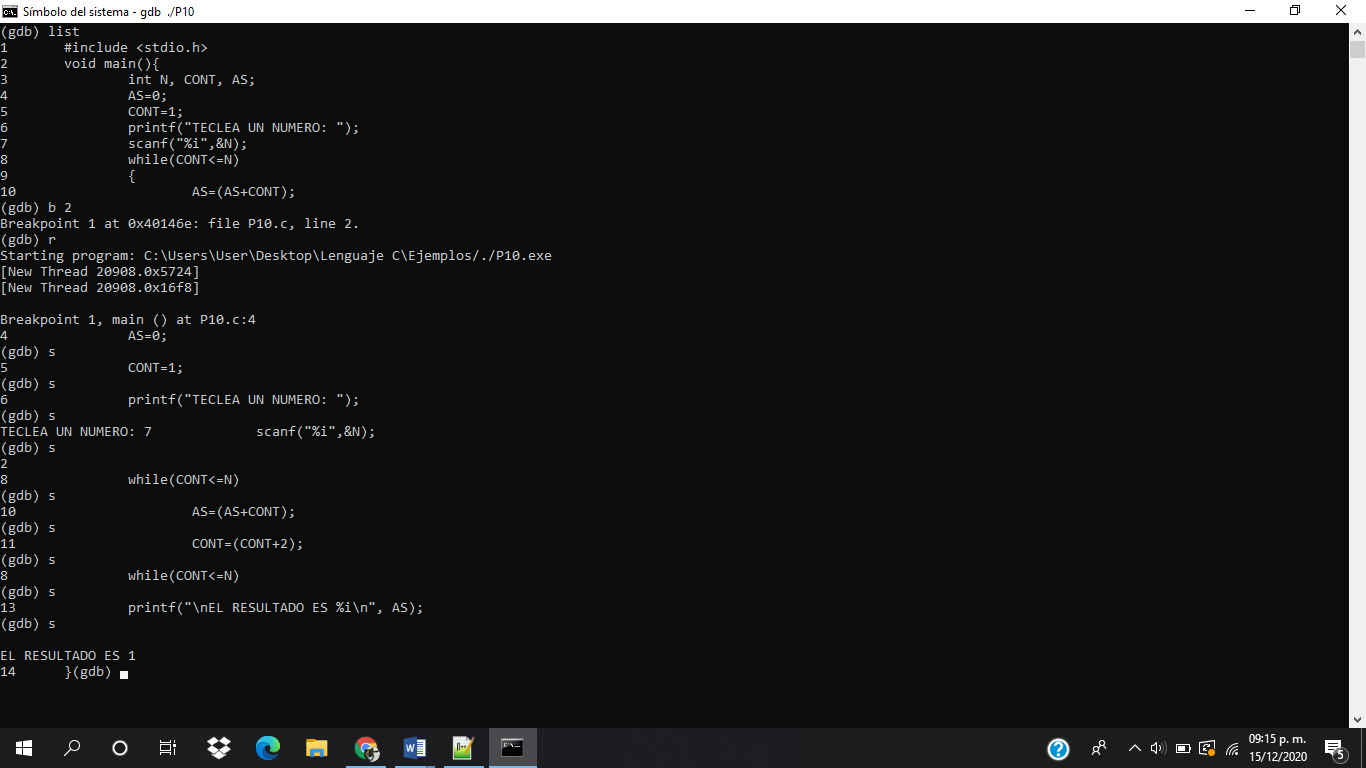






**Ejercicios propuestos**

**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.**



La condición se cumplió, siguió haciéndolo hasta que “CONT” se volvió más grande que “N”, el número dado, ya que si era menor o igual debía seguir ejecutándose.

**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++)

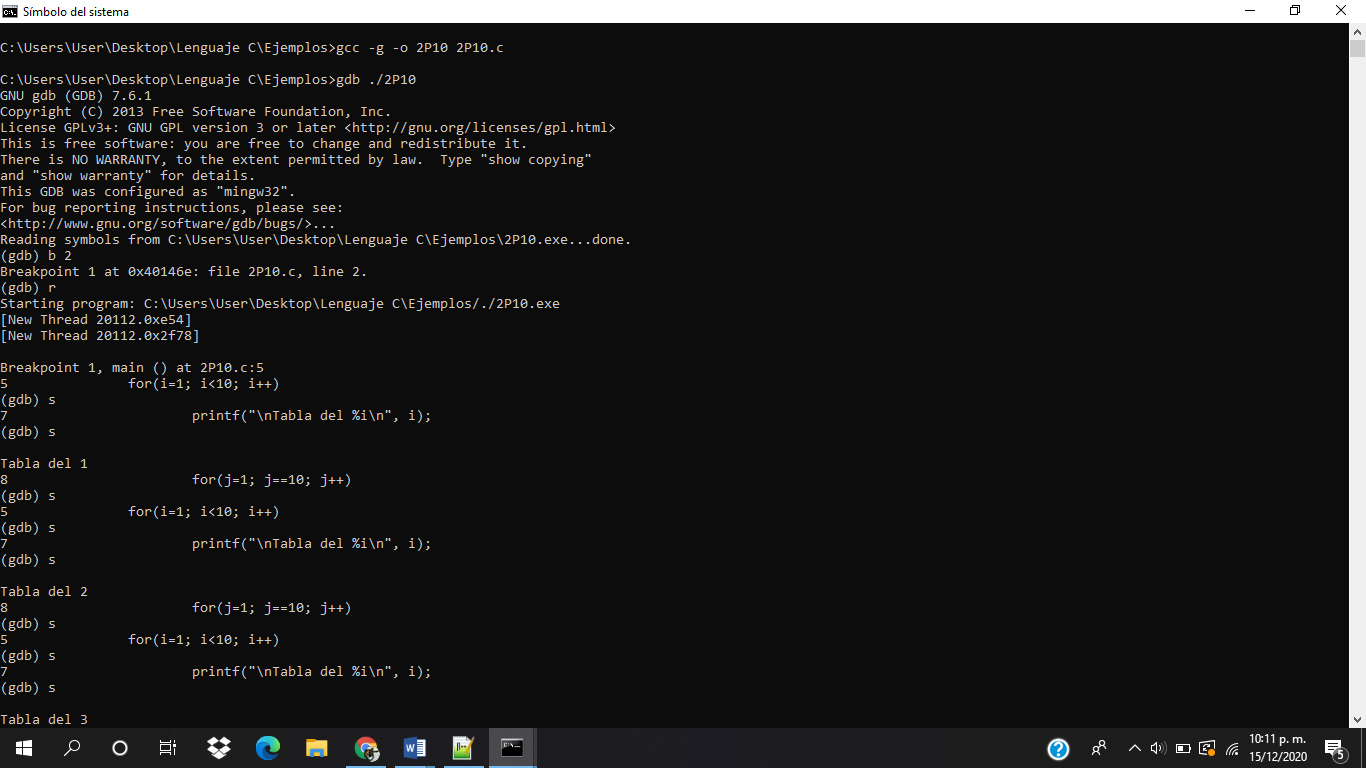
{

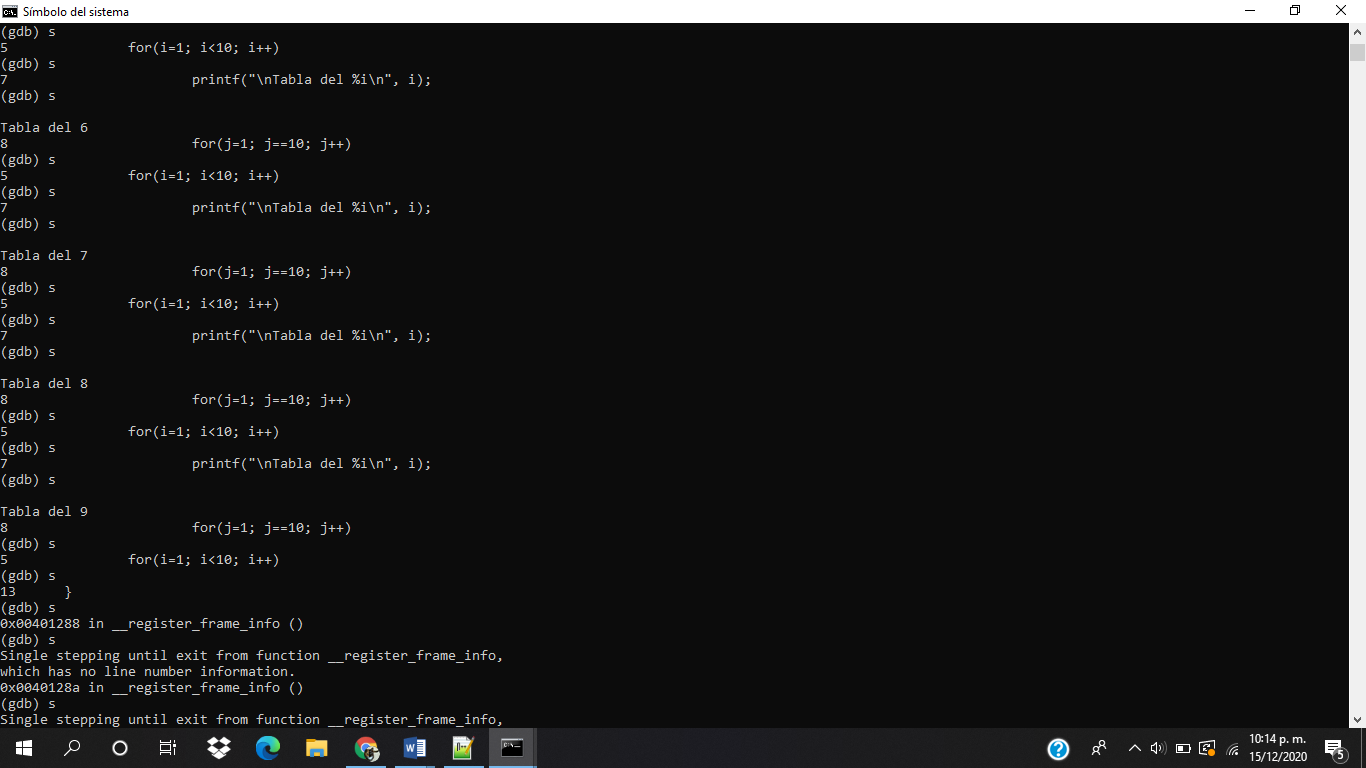
printf("%i X %i = %i\n", i, j, i\*j);

}

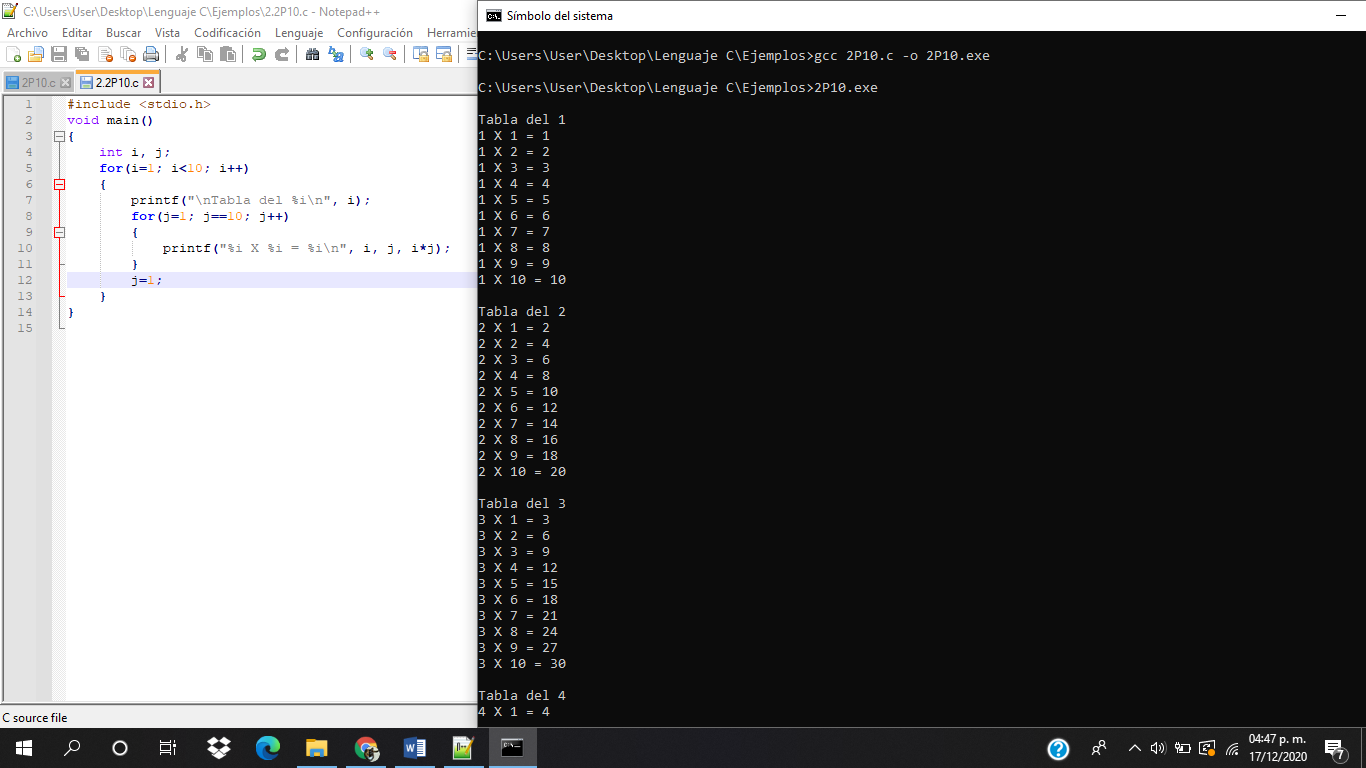
}

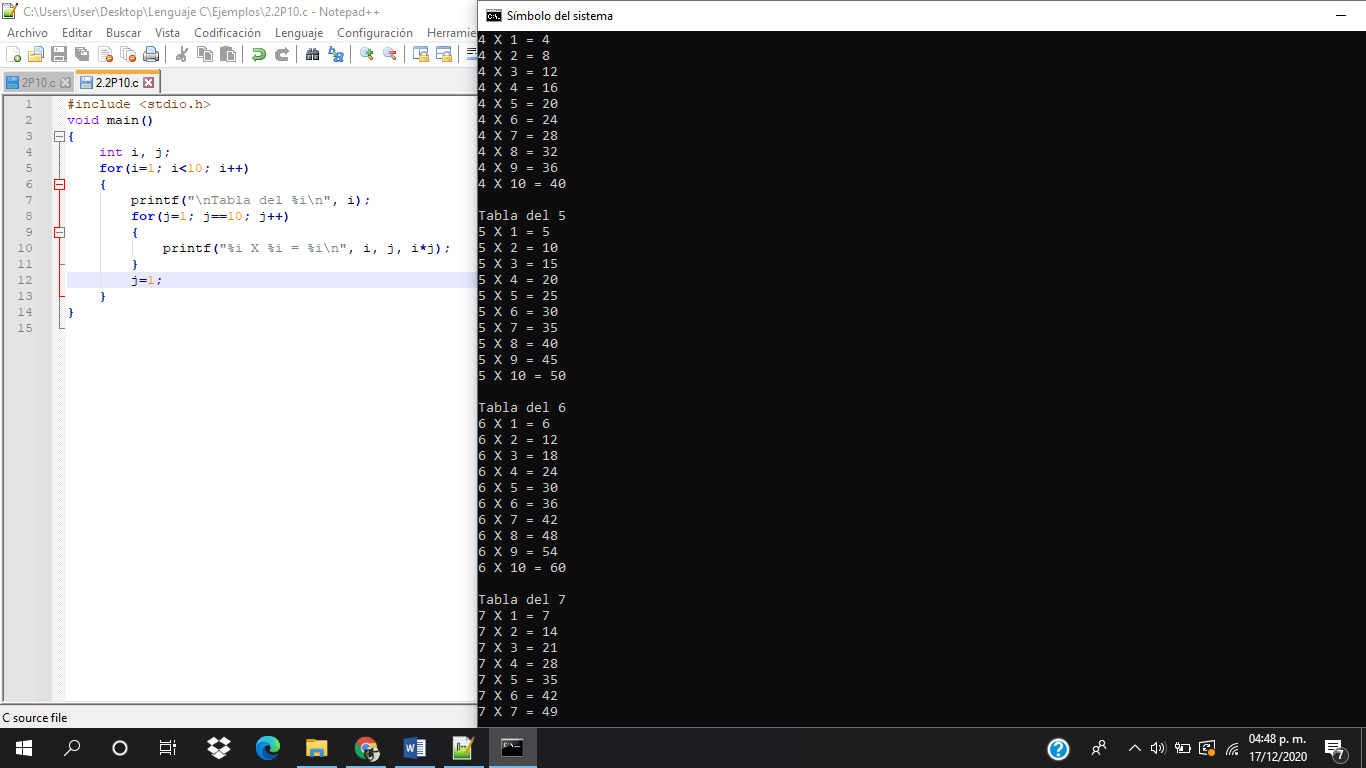
}

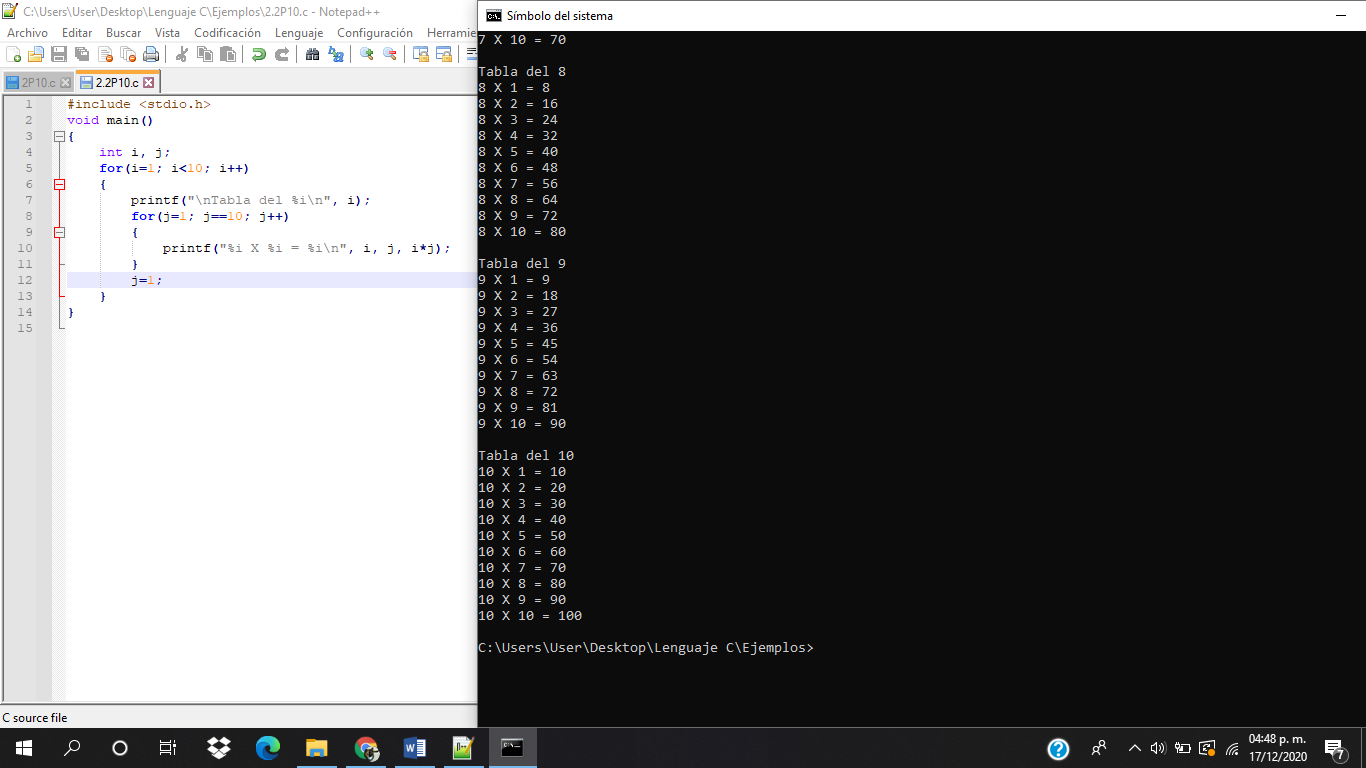


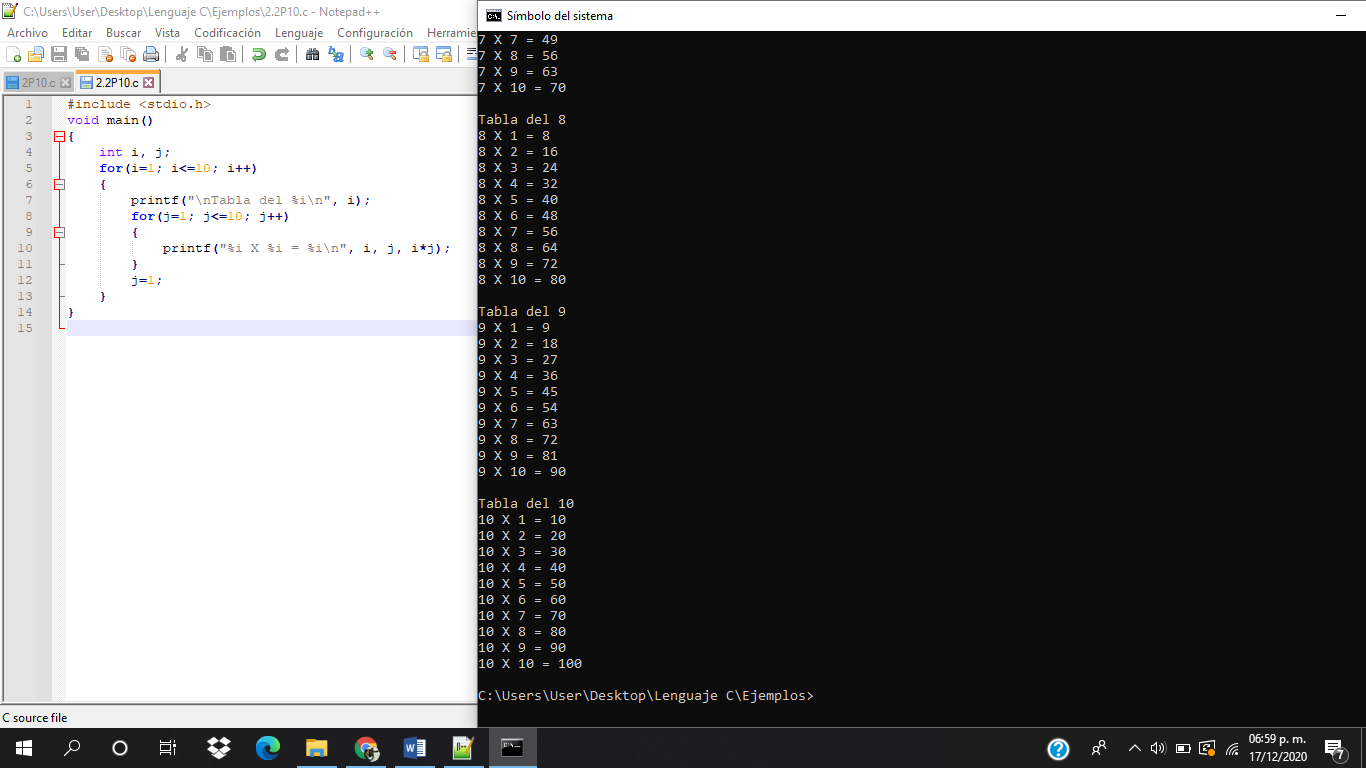
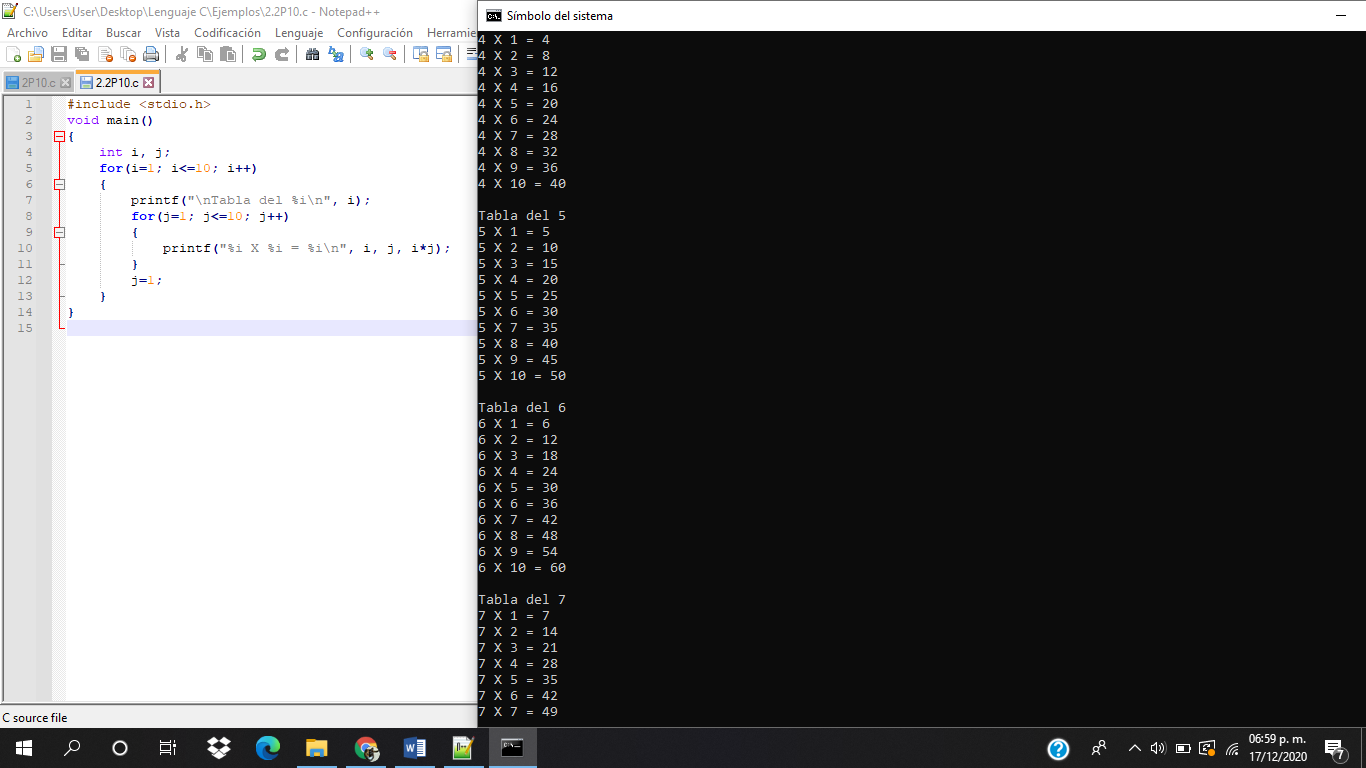
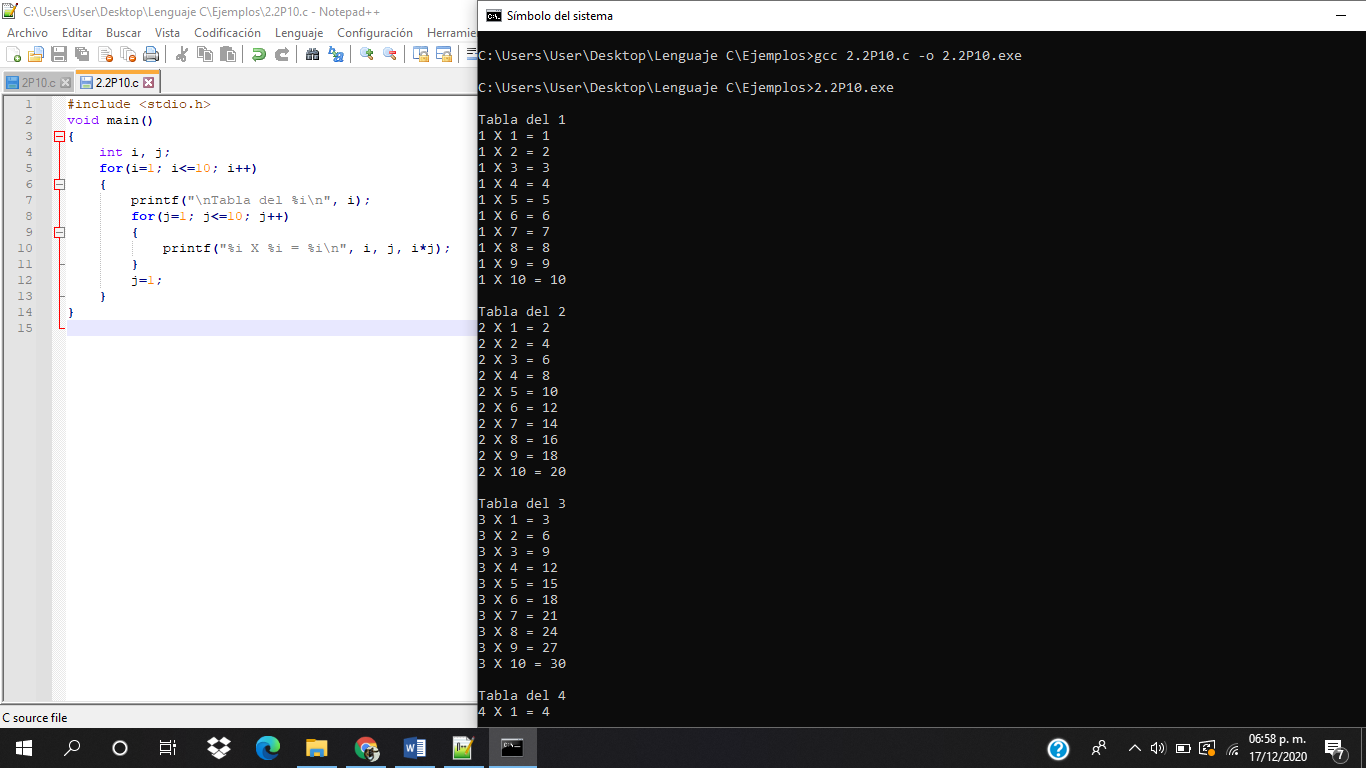


Aquí se observa cómo es que solo da los títulos el programa sin ninguna multiplicación







Primeramente hice una corrección utilizando do-while, lo que fue bastante largo el irlo probando, hasta que resulto en esto

Pero volví a ver el programa original, notando que los errores eran más las condicionales, ya que en ningún momento iba a llegar a ser igual a 10, por lo que no iba a ser verdadera la condicional y no haría la acción. Consecutivamente si no ponía nuevamente que j=1 se quedaría con j=11 y así aumentando de 1 en 1. Finalmente i debía tener i<=10 porque sino se quedaba en menor que 10, dando las multiplicaciones 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.**#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;

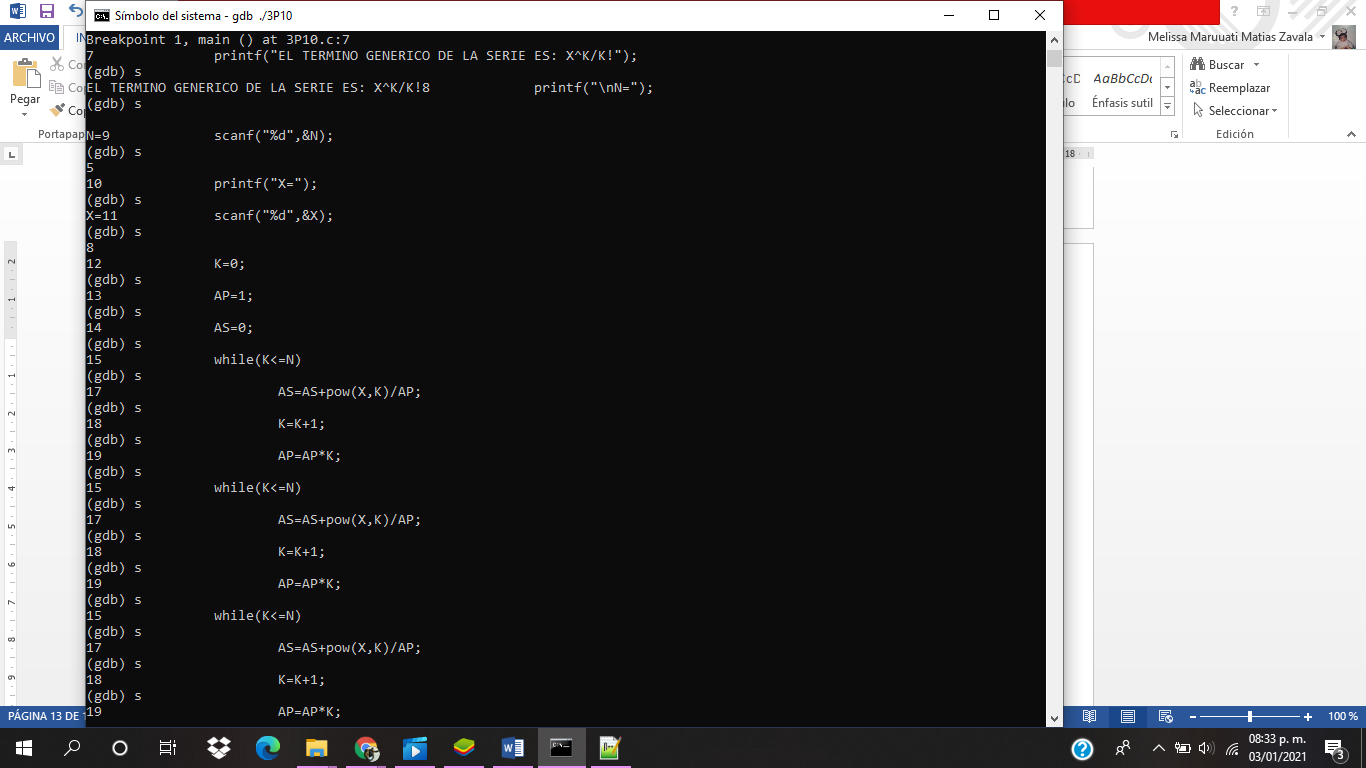
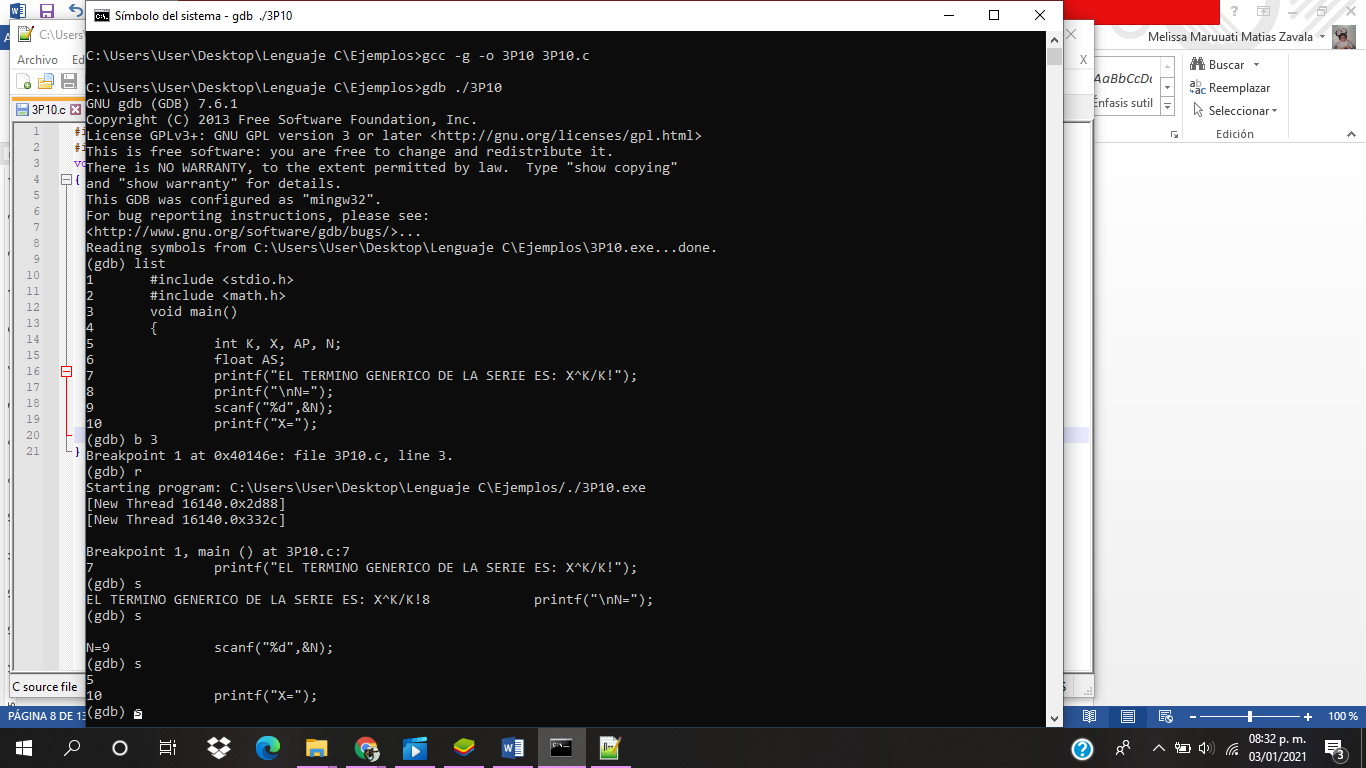
K=K+1;

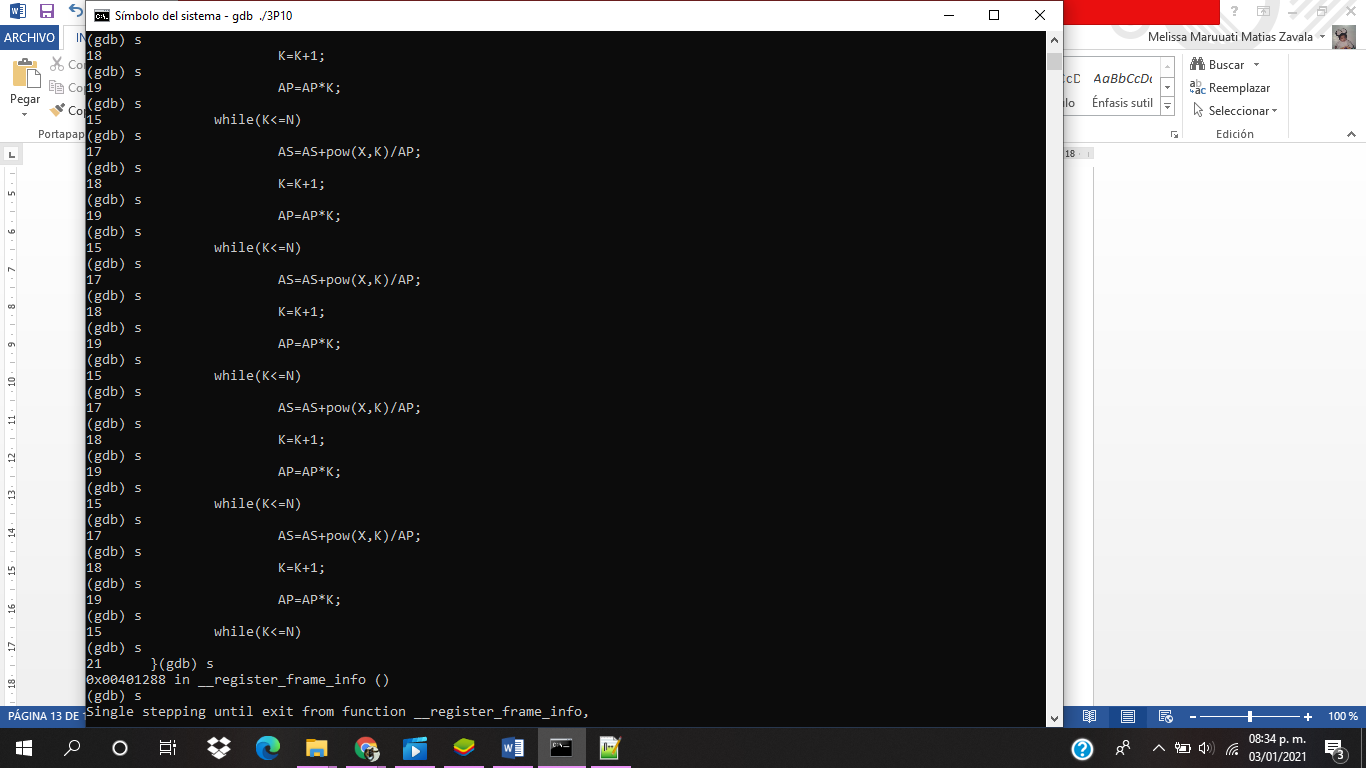
AP=AP\*K;

}

printf("SUM=%le",AS);

}





**Conclusión**

No comprendía como es que se empezaba a correr el programa, ya una vez después de analizarlo completamente vi que ya lo había hecho y a partir de ahí fui poniendo los diversos comandos para comprender la sintaxis con la que esta trabajaba.

Finalmente el último ejercicio no lo pude hacer por más que lo veía, ya que no comprendí su función, así como con eso no sabía cuál parte era el error representado en este

**Referencias**

*-Manual de prácticas del laboratorio de Fundamentos de programación*. Facultad de Ingeniería UNAM. Recuperado de: <http://lcp02.fi-b.unam.mx/poll/login/>

*-Curso Lenguaje en C. Soluciones MyL.* De <http://solucionesmyl.com/cursos/lenguaje_c/scanf.html>