|  |  |  |
| --- | --- | --- |
|  | **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):* | 9 |
| *Integrante(s):* | Camacho Garduño Miguel Angel |
| *No. de Equipo de cómputo empleado:* | No aplica |
| *No. de Lista o Brigada:* | 4 |
| *Semestre:* | 2021-1 |
| *Fecha de entrega:* | 6 de diciembre de 2020 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Guía práctica de estudio 09:

Estructuras de repetición

# Objetivo:

Elaborar programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva define.

# Actividades:

* Elaborar un programa que utilice la estructura while en la solución de un problema
* Elaborar un programa que requiera el uso de la estructura do.while para resolver un problema. Hacer la comparación con el programa anterior para distinguir las diferencias de operación entre while y do-while
* Resolver un problema dado por el profesor que utilice la estructura for en lugar de la estructura while
* Usa la directiva define para elaboración de código versátil.

# Introducción:

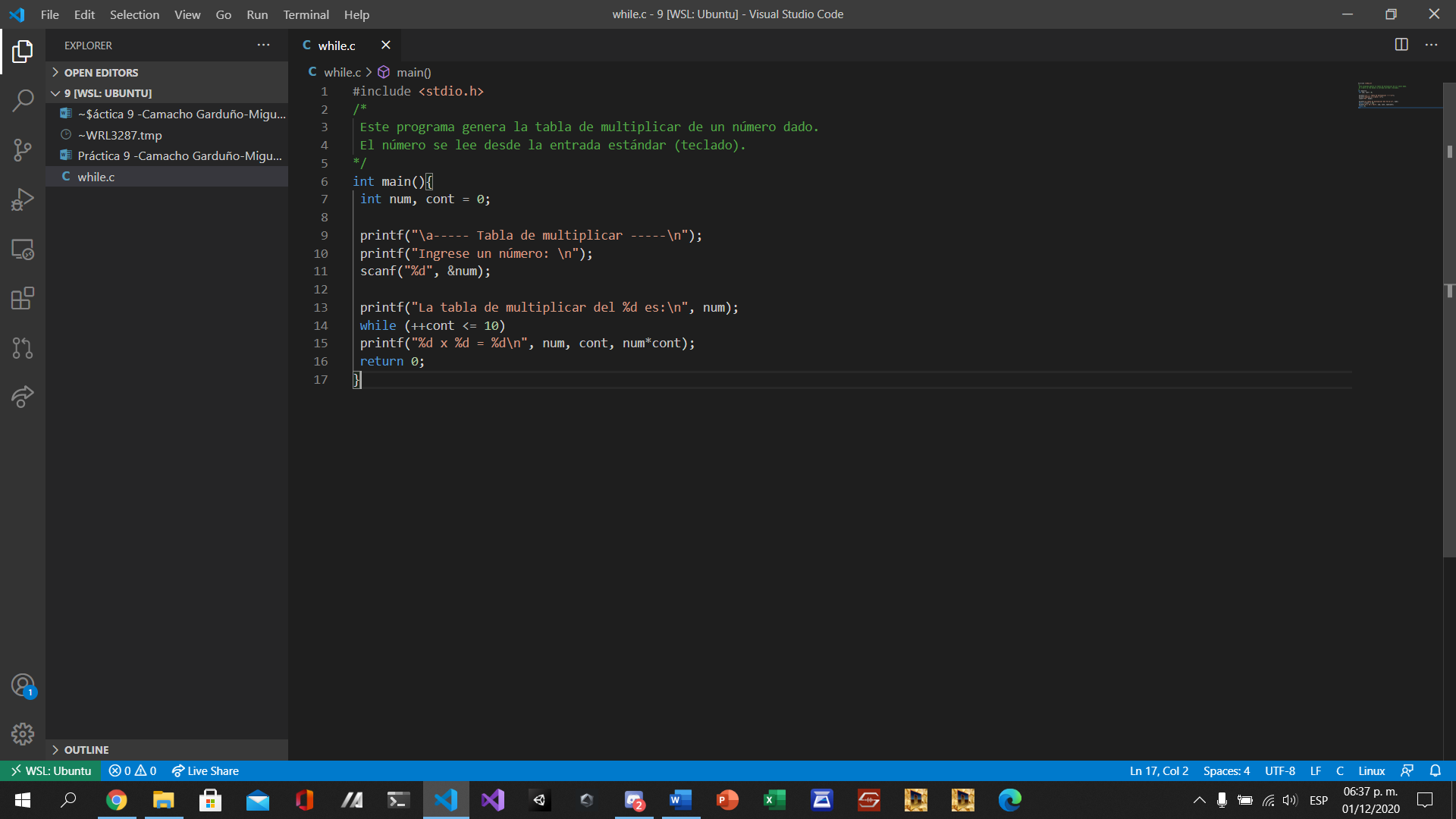
Las estructuras de repetición permiten ejecutar un conjunto de instrucciones de manera repetida mientras que la expresión lógica a evaluar se cumpla

La estructura repetitiva (o iterativa) while primero valida la expresión lógica y si ésta se cumple (es verdadera) procede a ejecutar el bloque de instrucciones de la estructura, el cual está delimitado por las llaves {}. Si la condición no se cumple se continúa el flujo normal del programa sin ejecutar el bloque de la estructura, es decir, el bloque se puede ejecutar de cero a ene veces. do-while es una estructura cíclica que ejecuta el bloque de código que se encuentra dentro de las llaves y después valida la condición, es decir, el bloque de código se ejecuta de una a ene veces. “for” permite realizar repeticiones cuando se conoce el número de elementos que se quiere recorrer. Las líneas de código que empiezan con # son directivas del preprocesador, el cual se encarga de realizar modificaciones en el texto del código fuente, como reemplazar un símbolo definido con #define por un parámetro o texto, o incluir un archivo en otro archivo con #include. Algunas veces es conveniente tener la posibilidad de abandonar un ciclo. La proposición break proporciona una salida anticipada dentro de una estructura de repetición, tal como lo hace en un switch. Un break provoca que el ciclo que lo encierra termine inmediatamente. La proposición continue provoca que inicie la siguiente iteración del ciclo de repetición que la contiene.

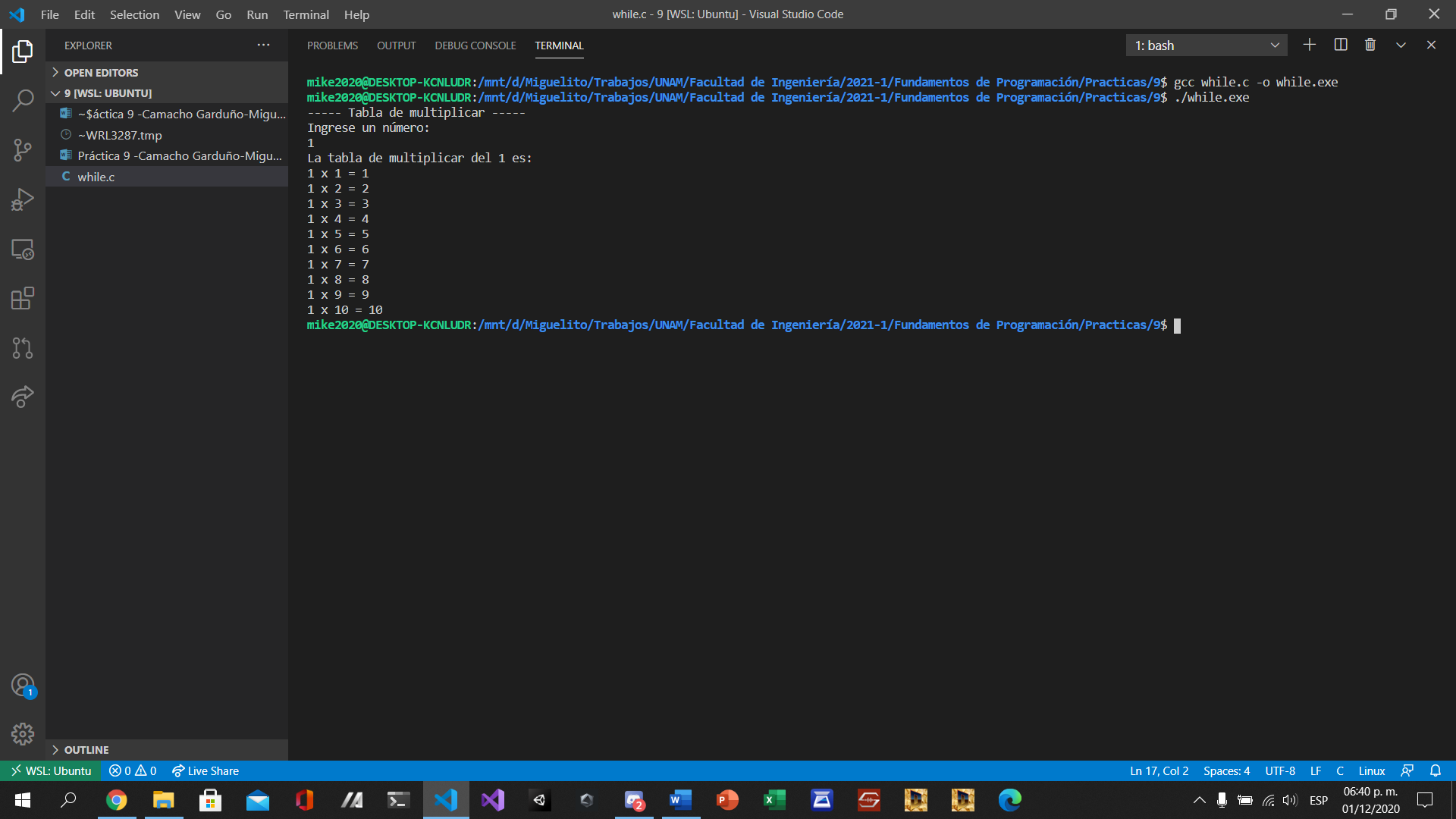
# Actividades

## Ejemplos del manual de prácticas

### While.c

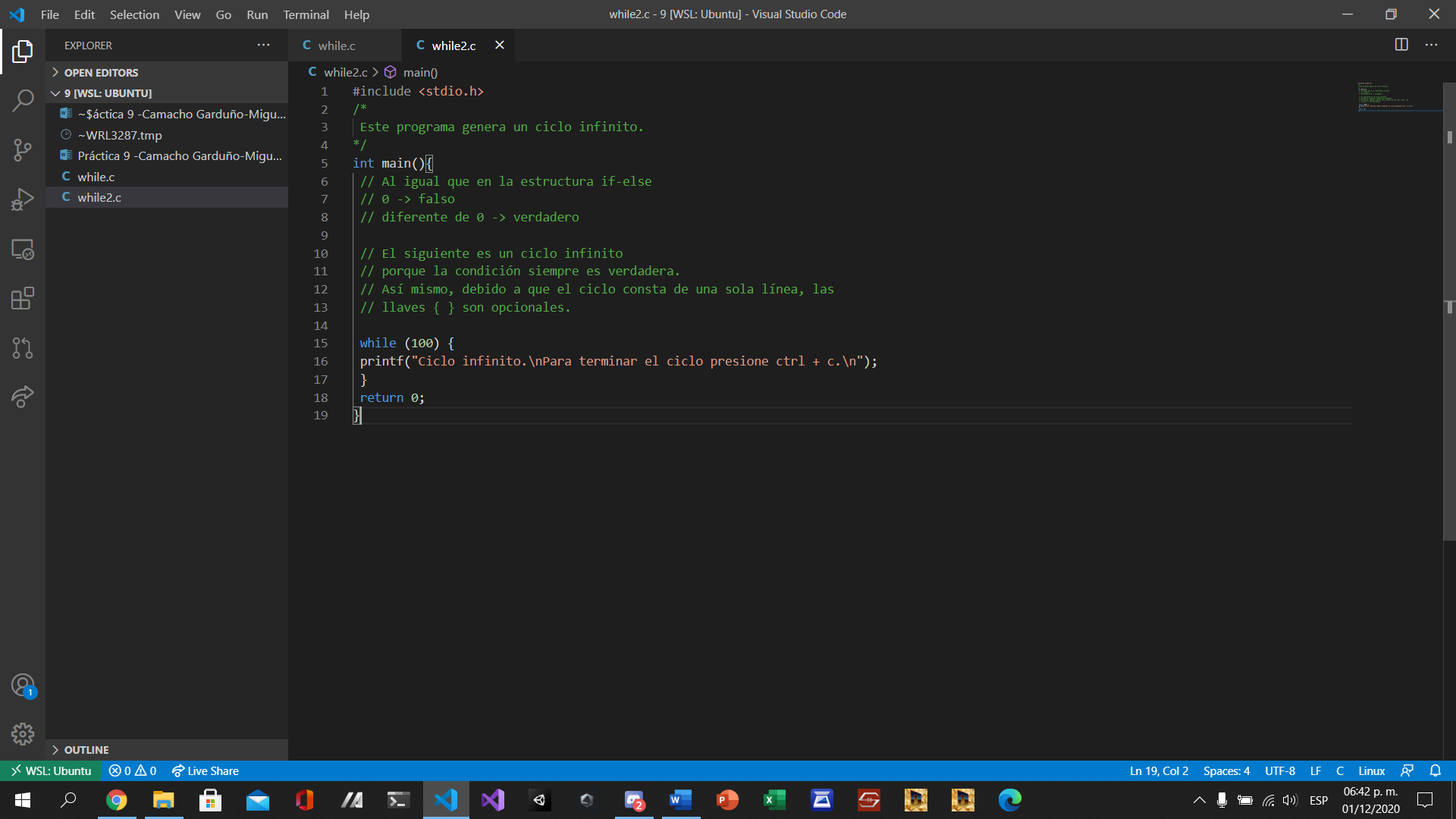


En una conexión remota a una terminal de Ubuntu en Visual Studio Code, transcribí el primer ejemplo de while

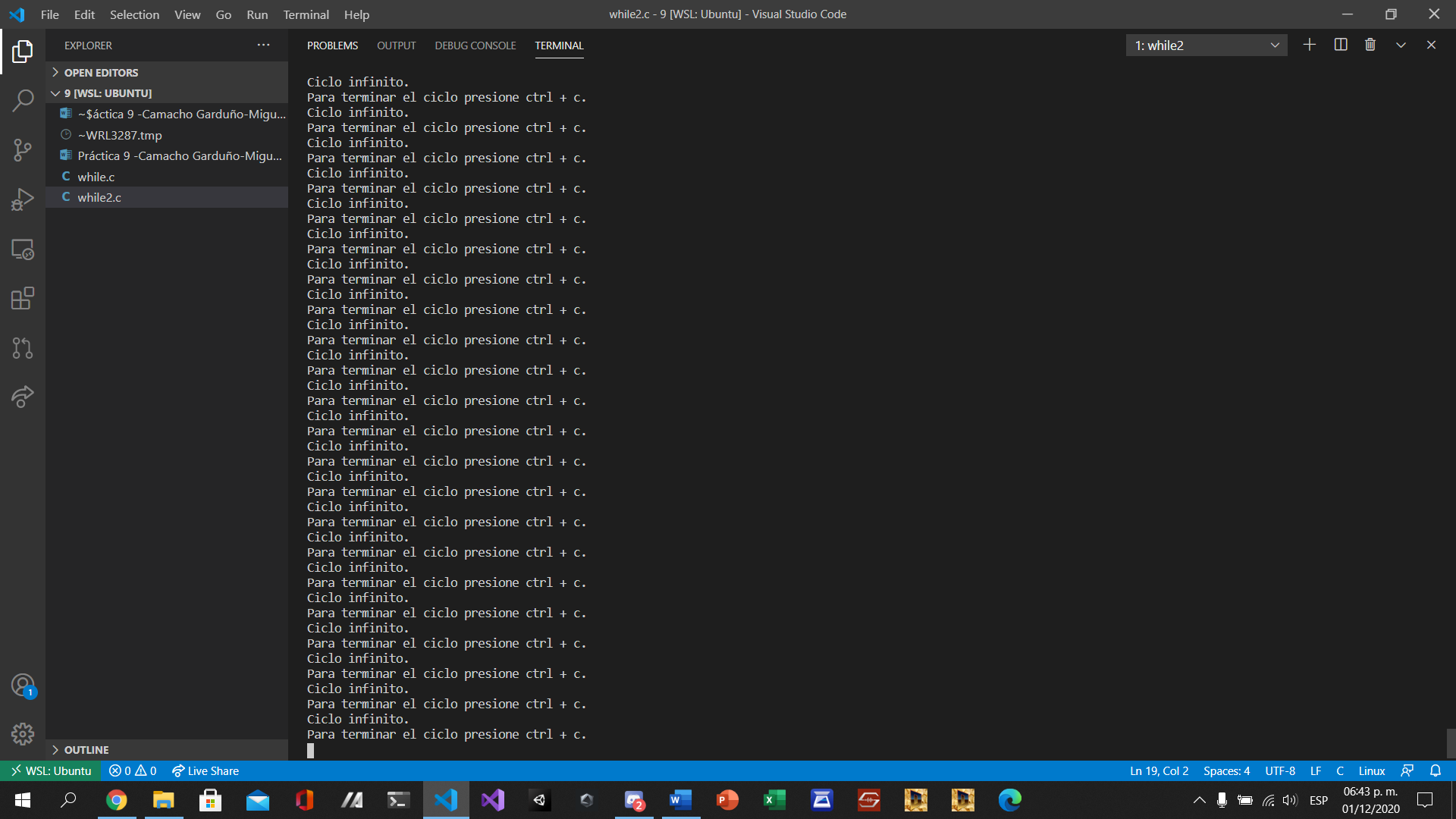


Al compilar y ejecutar el programa, me pide un número y me da los 10 valores del 1 de la tabla de multiplicar.

### While2.c

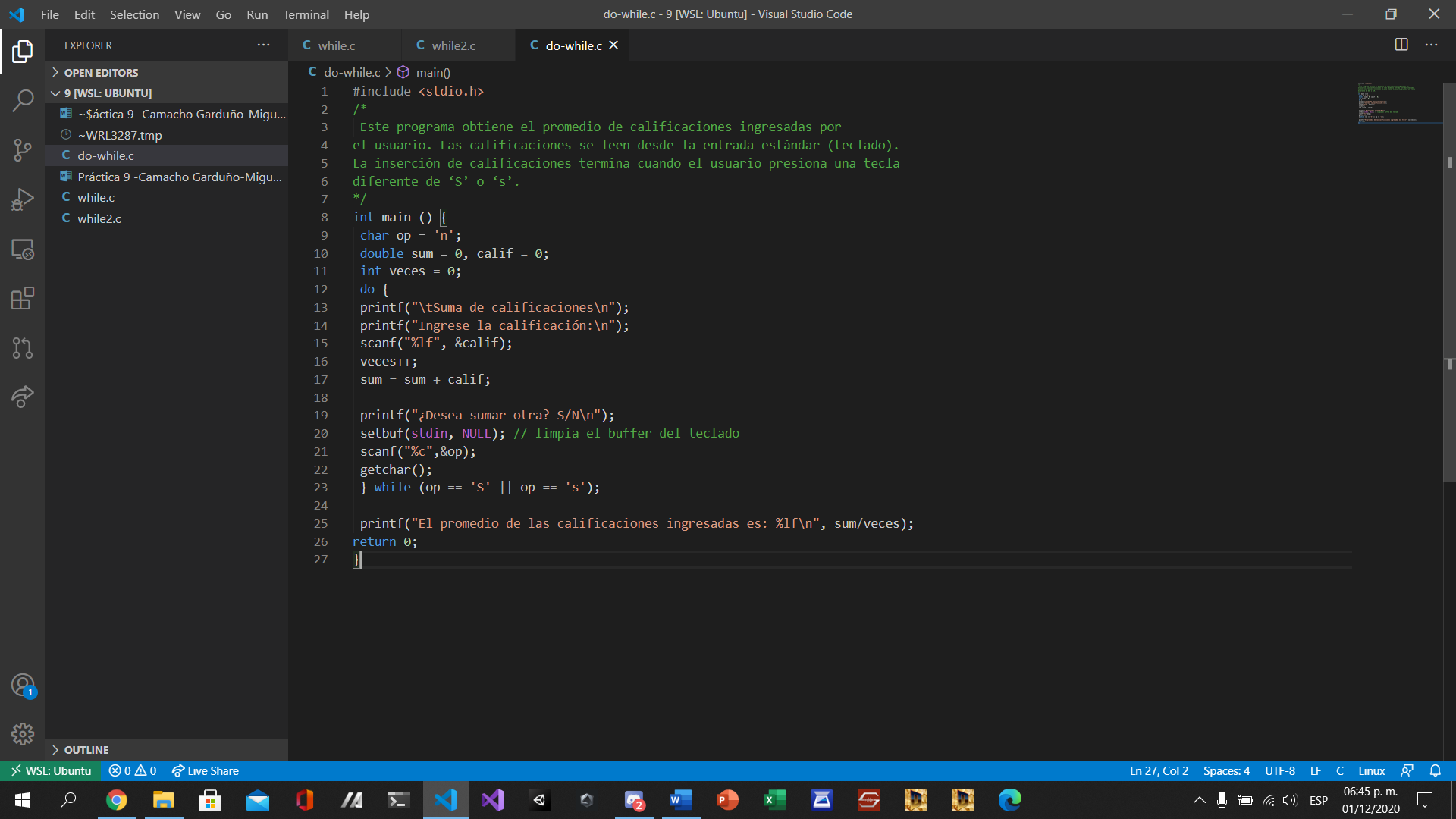


Transcribí el segundo ejemplo de while

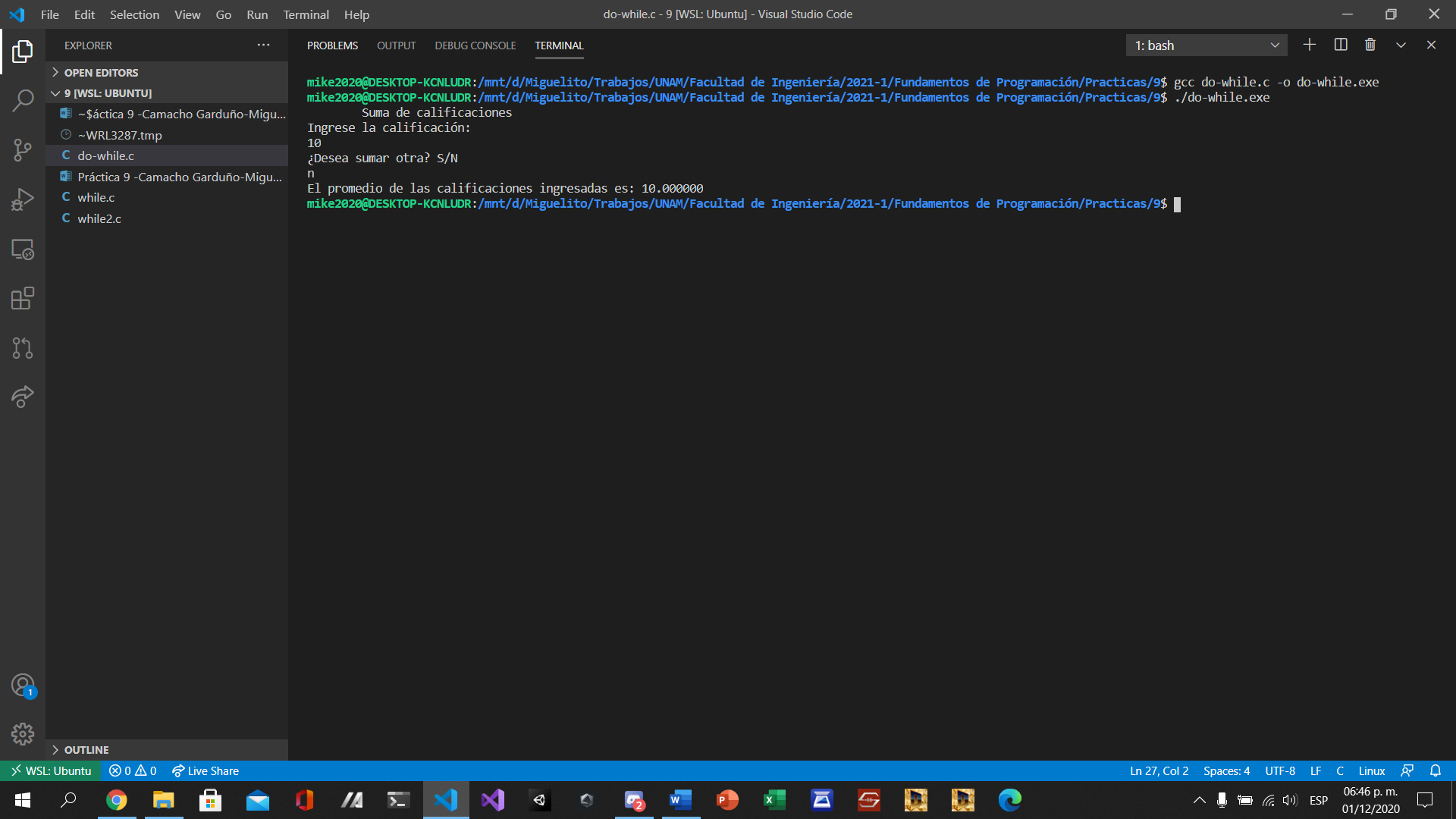


Al compilar y ejecutar, aparece infinitamente el mismo mensaje y que se detiene al presionar ctrl+c

### Do-while.c

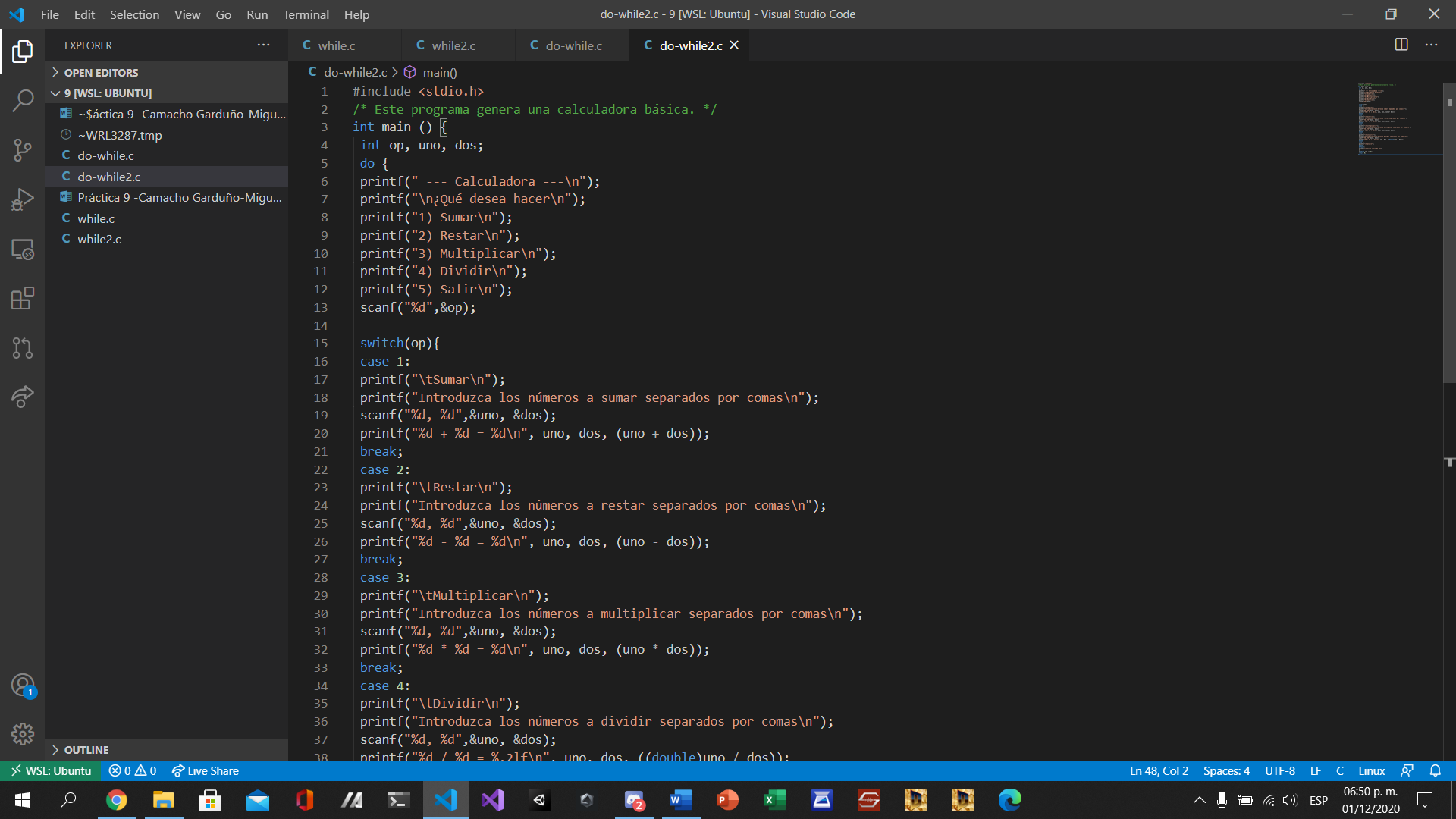


Transcribí el primer ejemplo de do-while

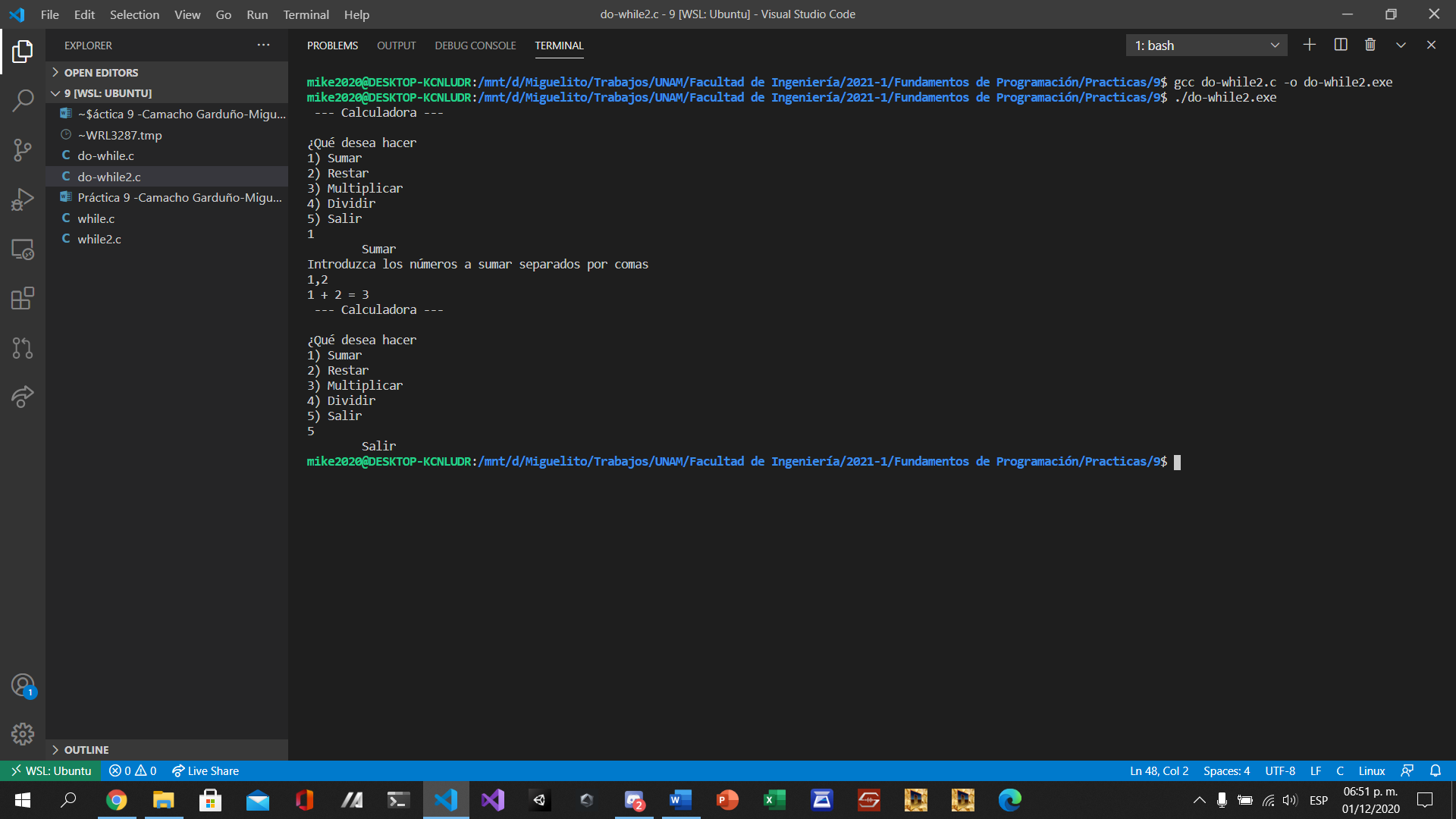


Al ejecutar el código, me pide dar una calificación y me pregunta si quiero sumar otra, al darle no, me da 10 de calificación total.

### Do-while2.c

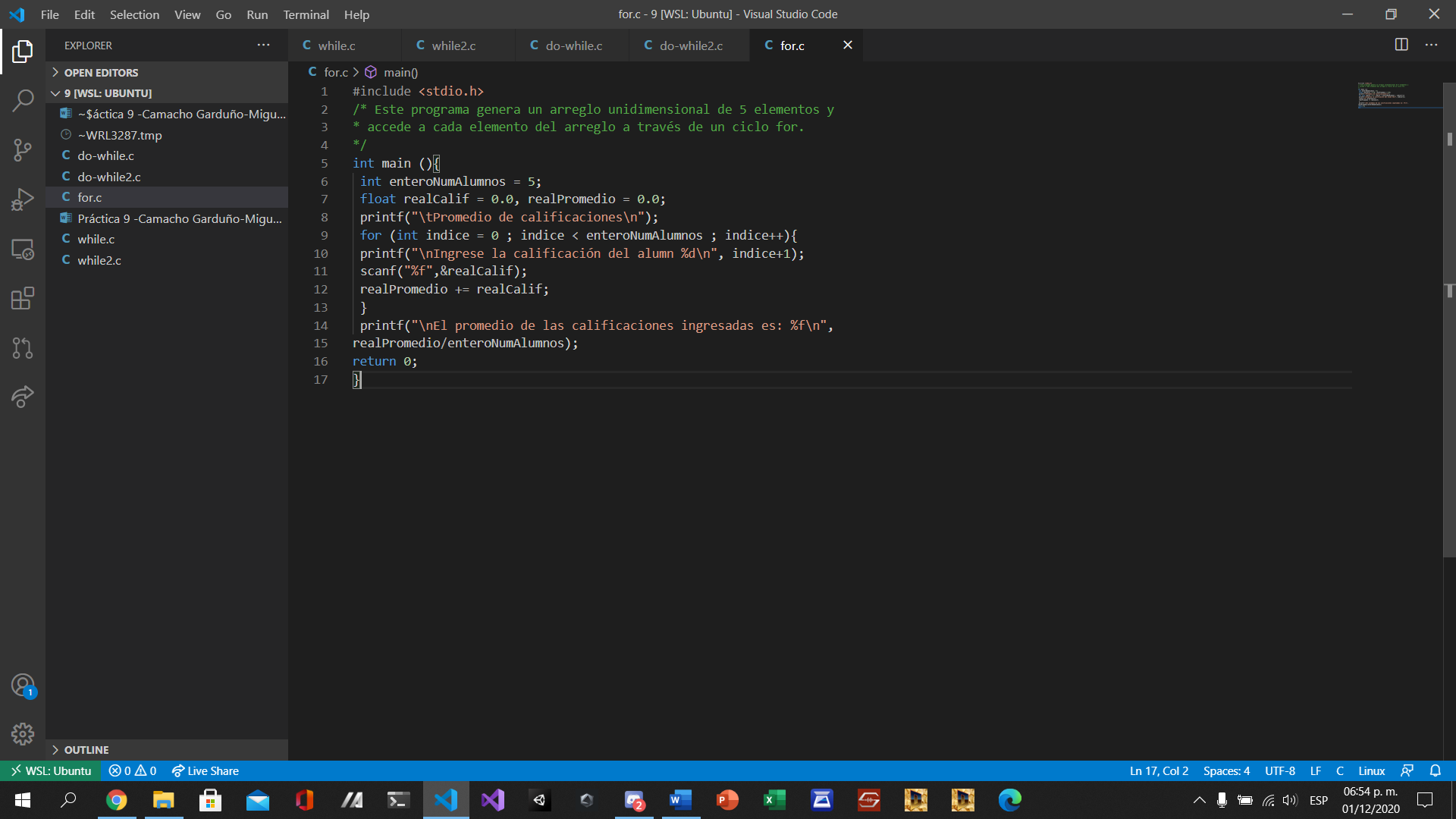


Transcribí el segundo ejemplo de do-while

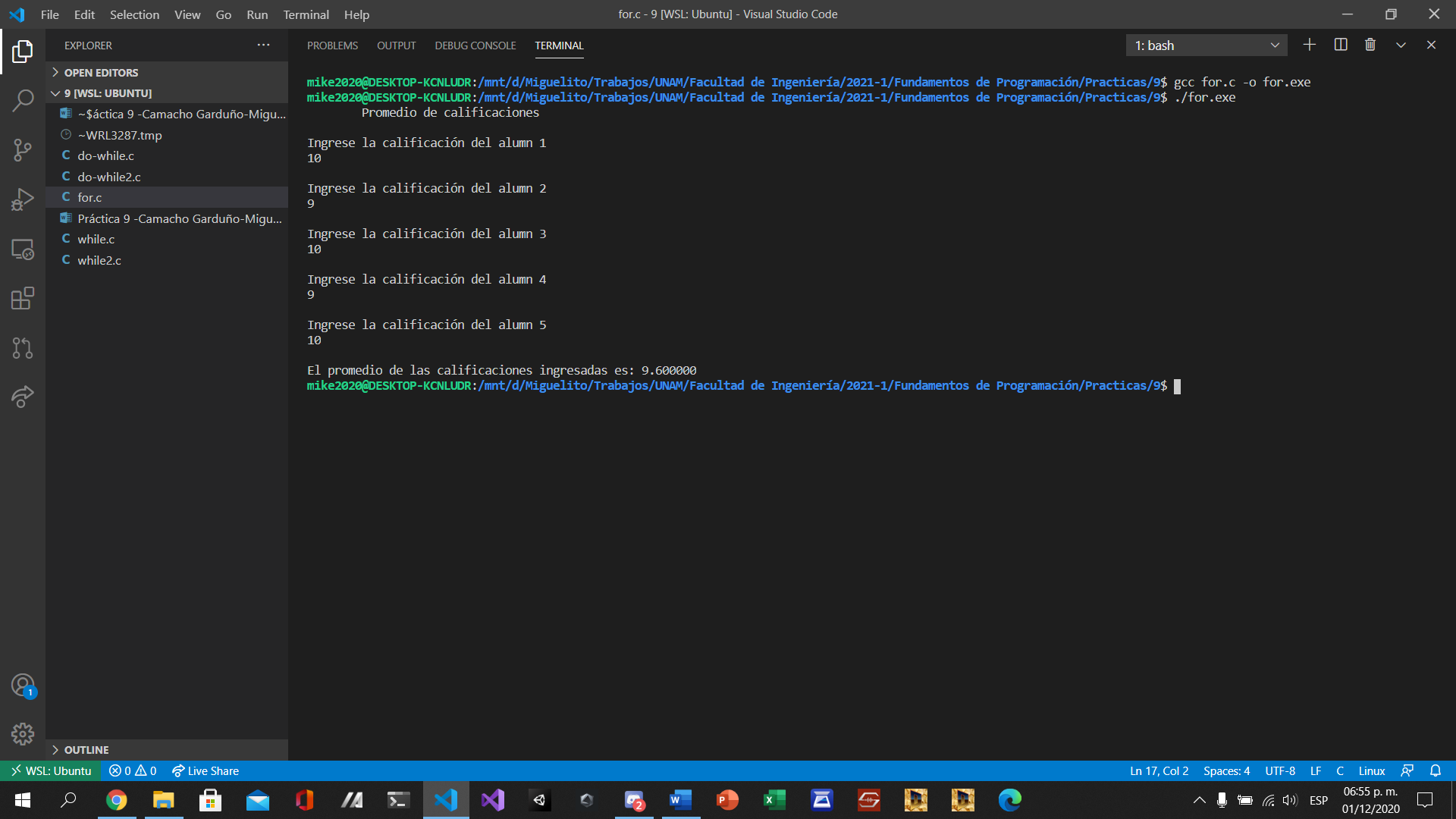


Al ejecutar el código, me pide seleccionar una opción, al darle suma me pide ingresar 2 número, al terminar, me regresa al menú principal y me pregunta que quiero hacer y le di en salí.

### For.c



Transcribí el ejemplo de for

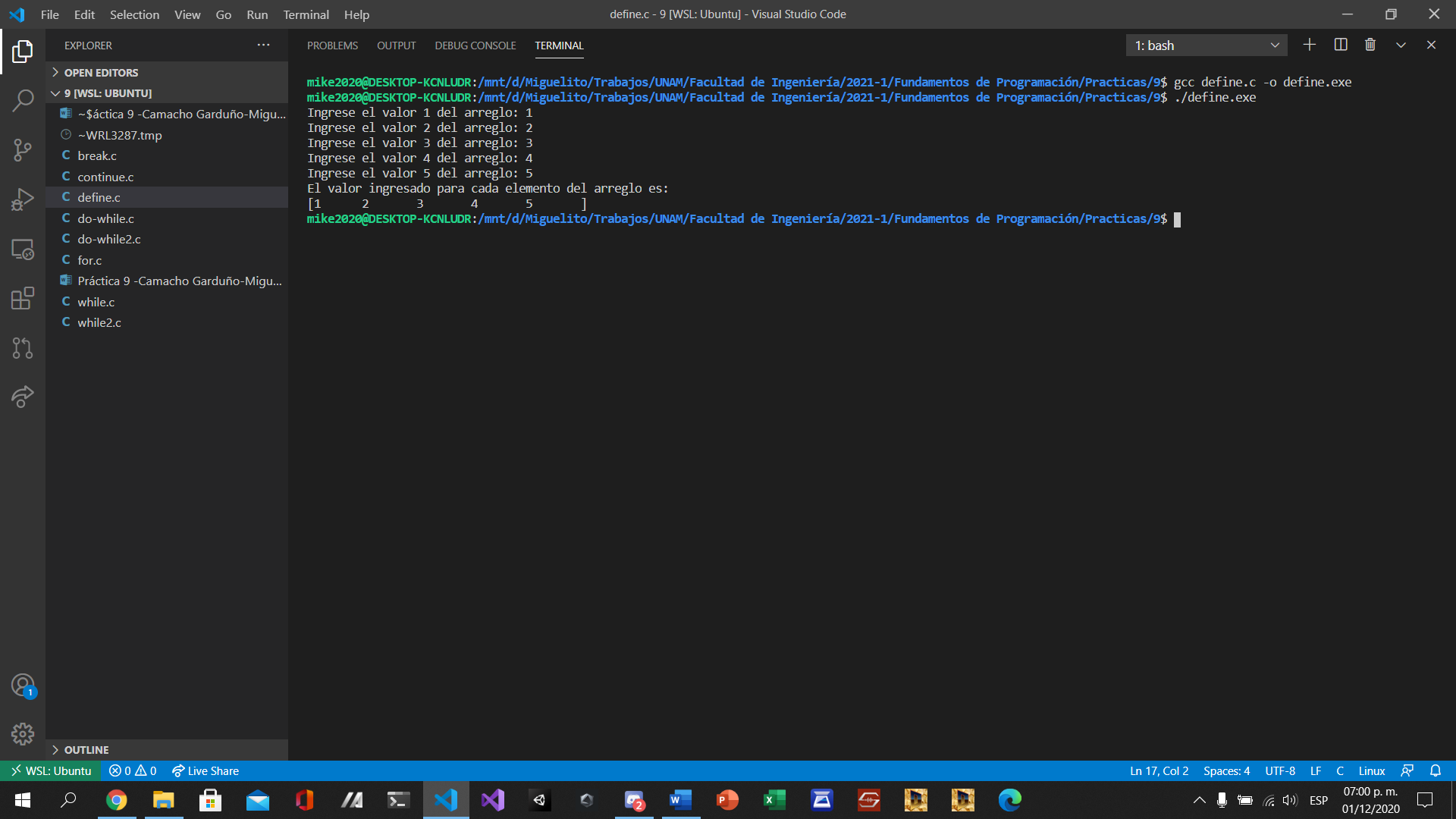


Al compilar y ejecutar el programa, me pide ingresar 5 calificaciones y me da el promedio de ellas.

### Define.c

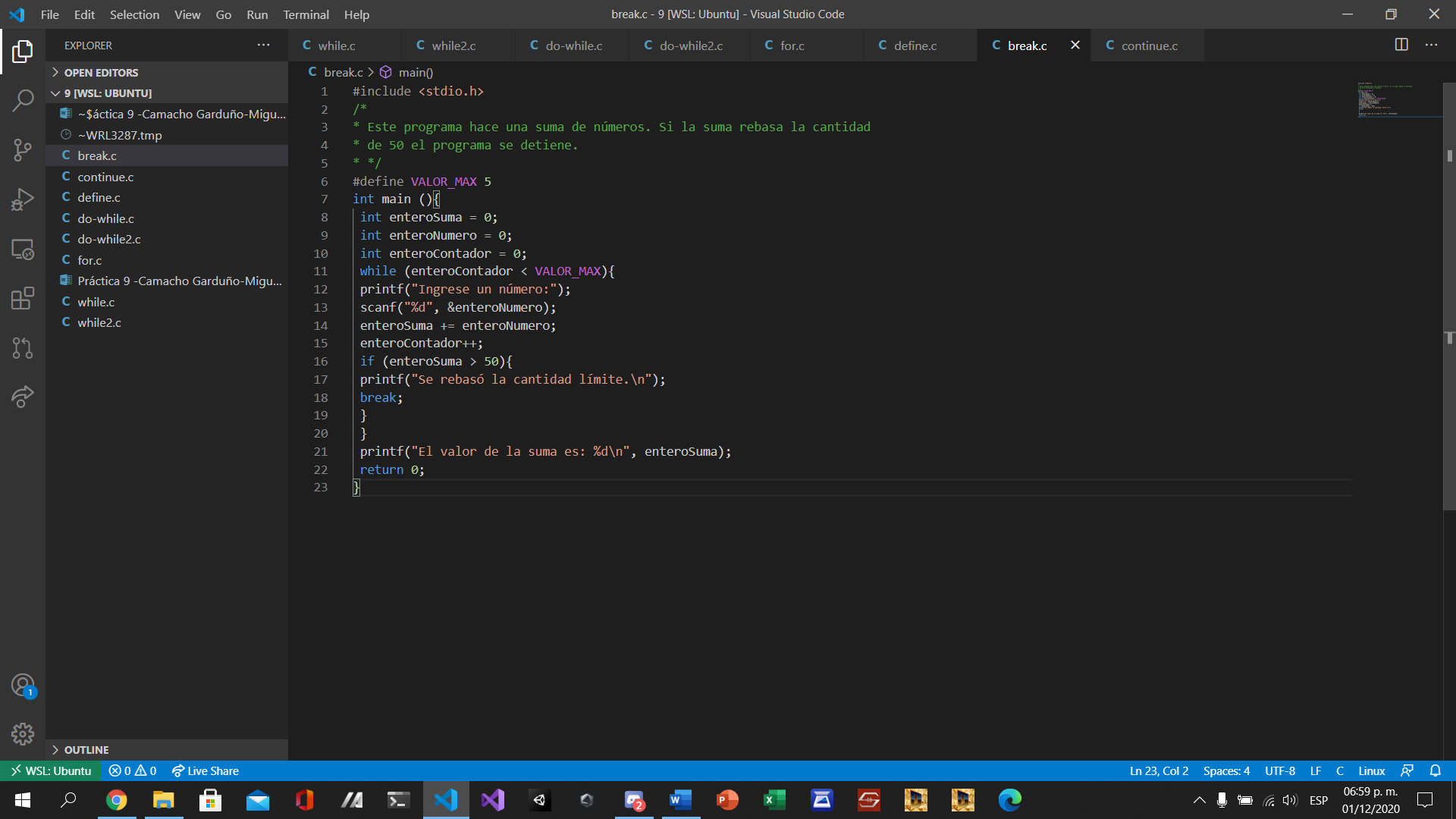


Transcribí el ejemplo de define

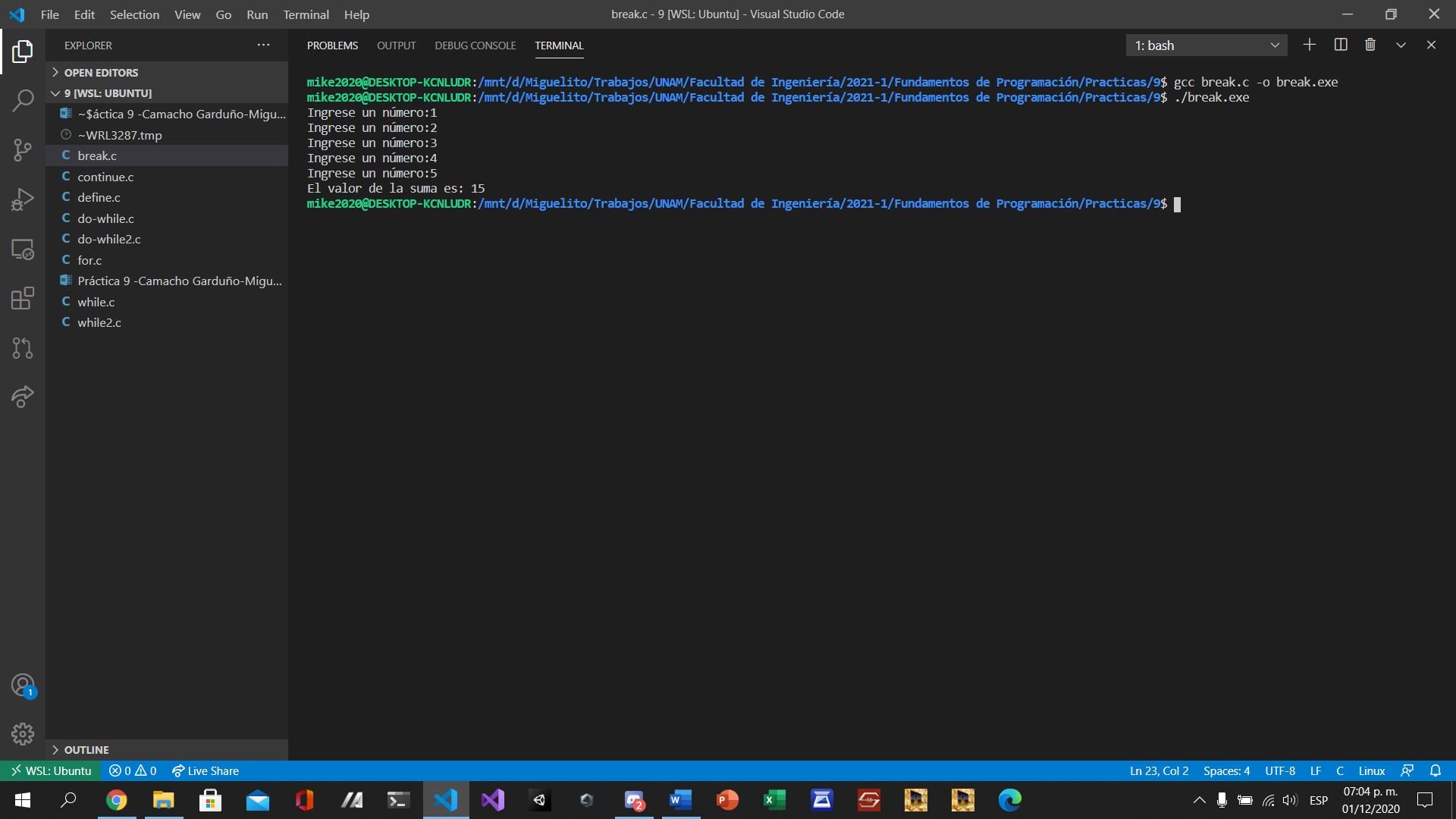


Al compilar y ejecutar el programa, me pide ingresar 5 arreglos y me da los 5 arreglas separados por espacios.

### Break.c

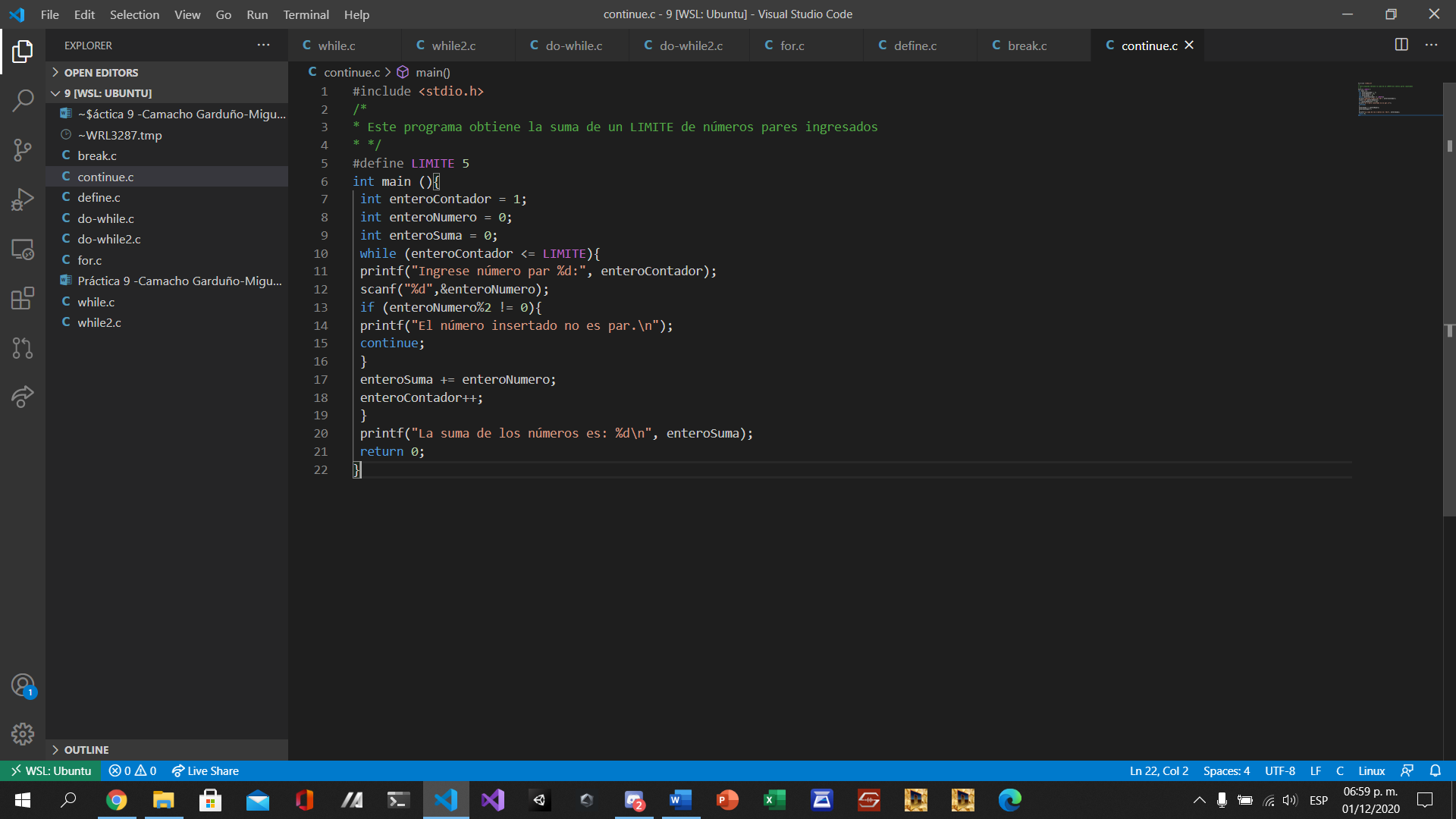


Transcribí el ejemplo de break

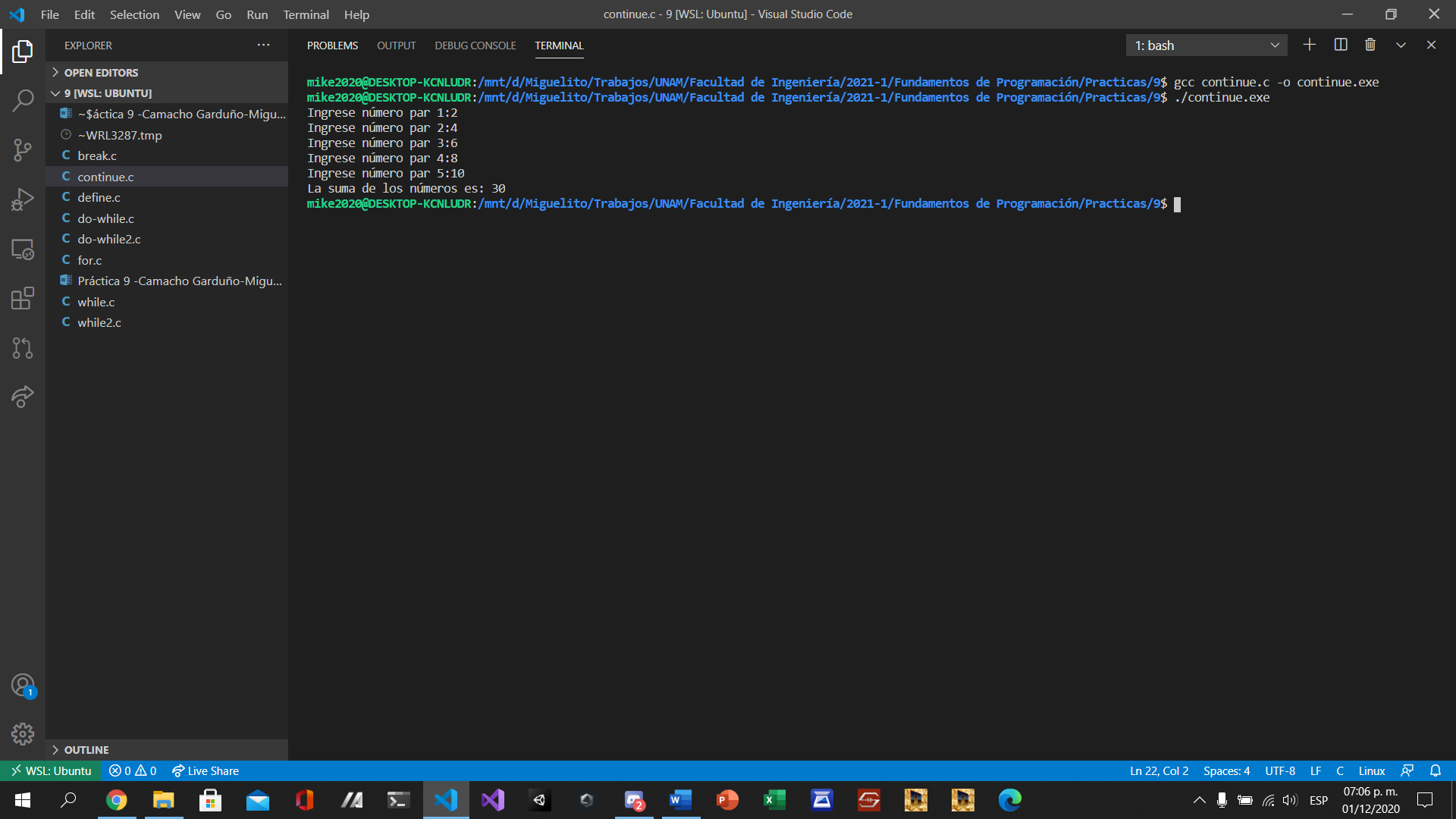


Al compilar y ejecutar el programa, me pide ingresar 5 número y me da la suma de esos 5 números

### Continue.c



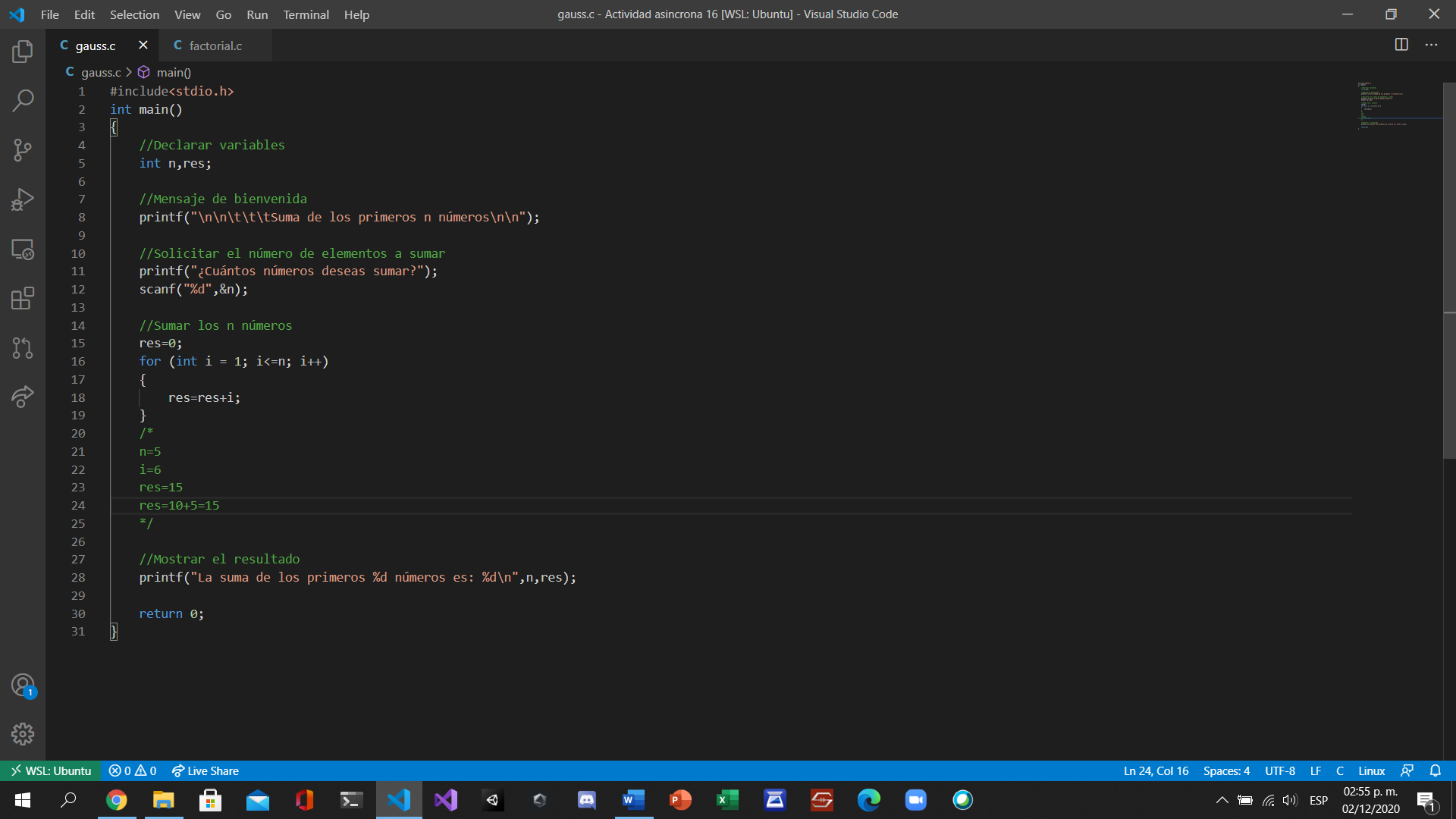
Transcribí el ejemplo de continue

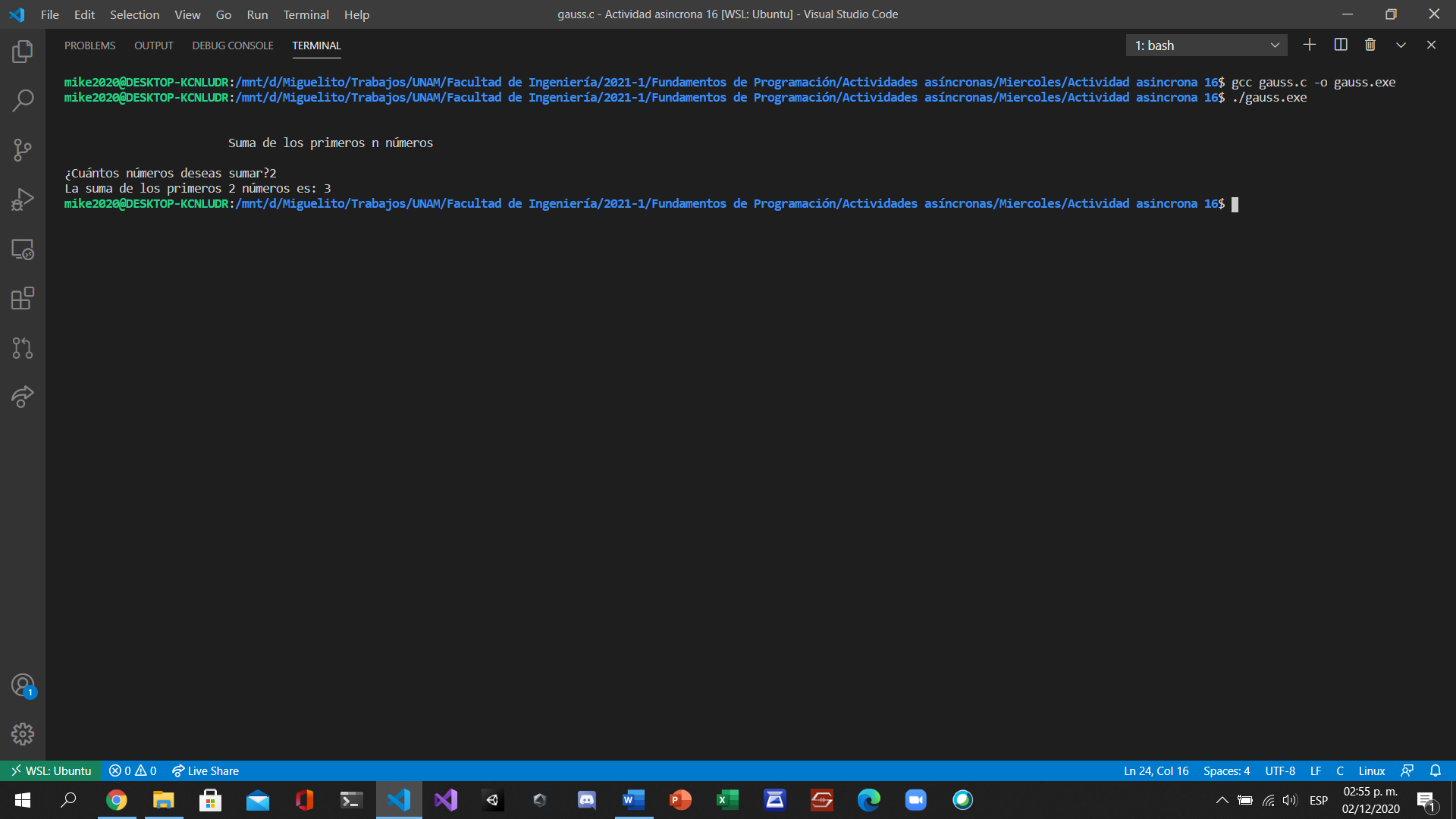


Al compilar y ejecutar un programa, me pide ingresar 5 números pares y me da la suma de esos 5 números

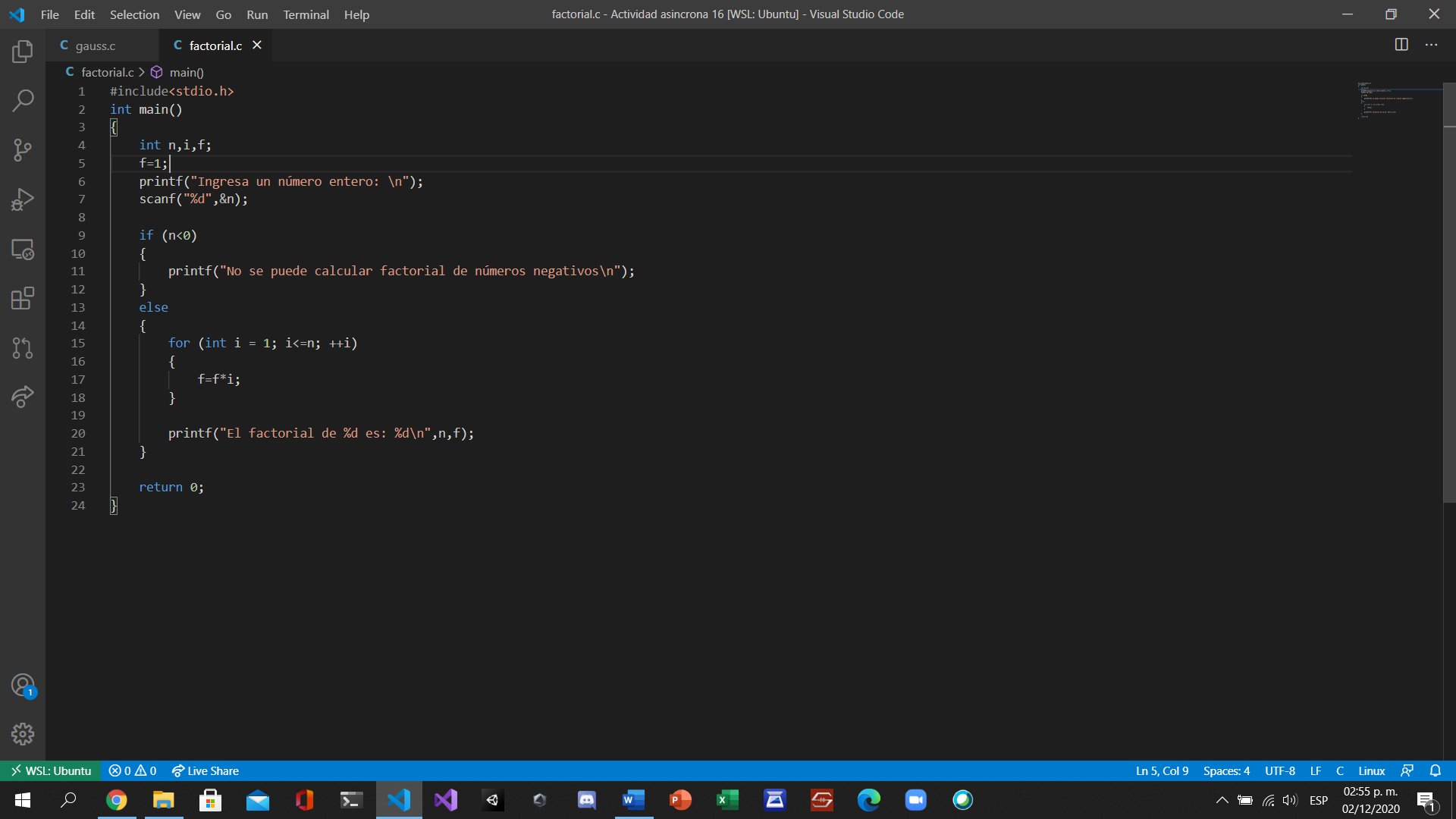
## Actividades dadas por el profesor

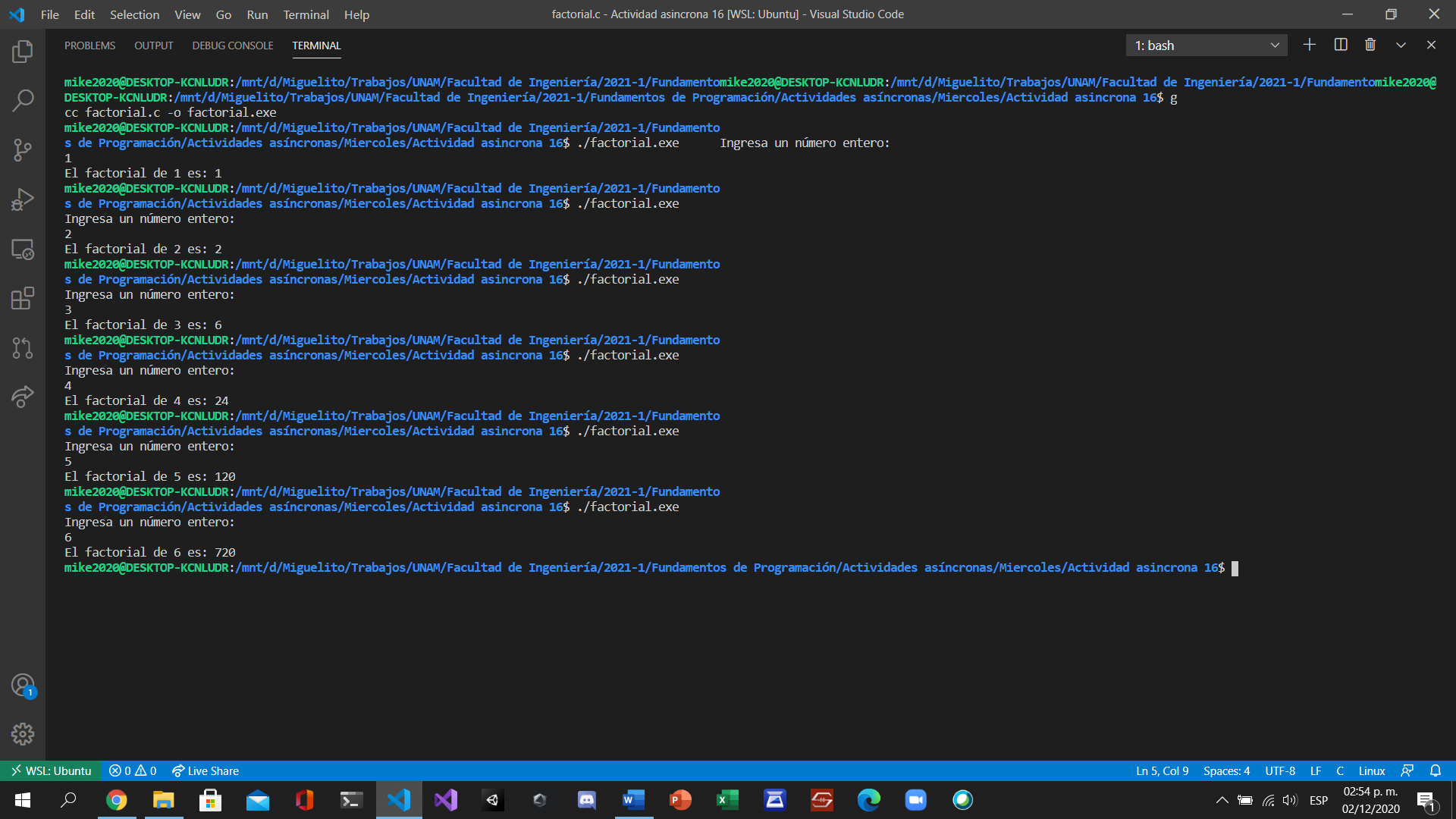
### Gauss.c



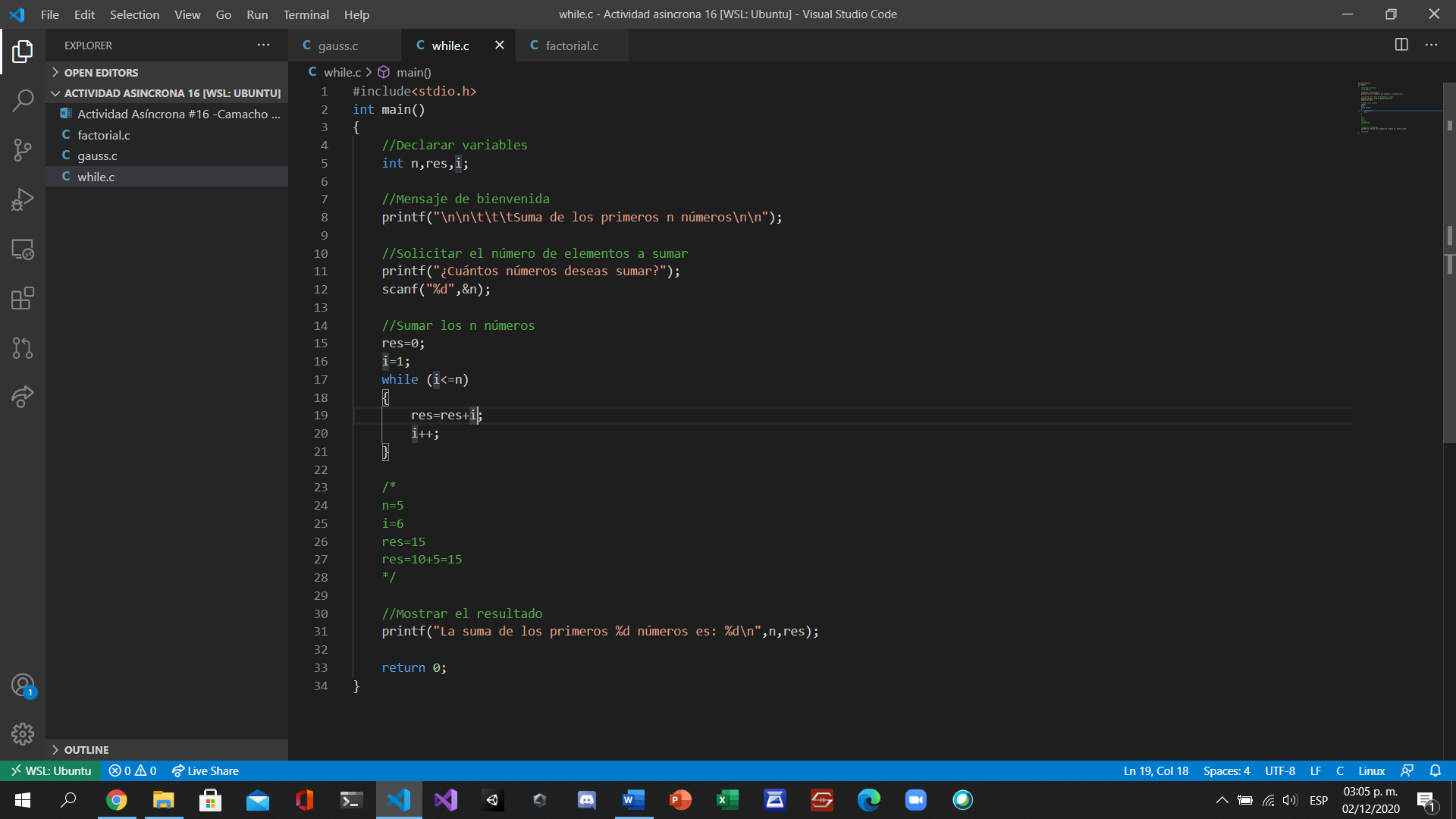


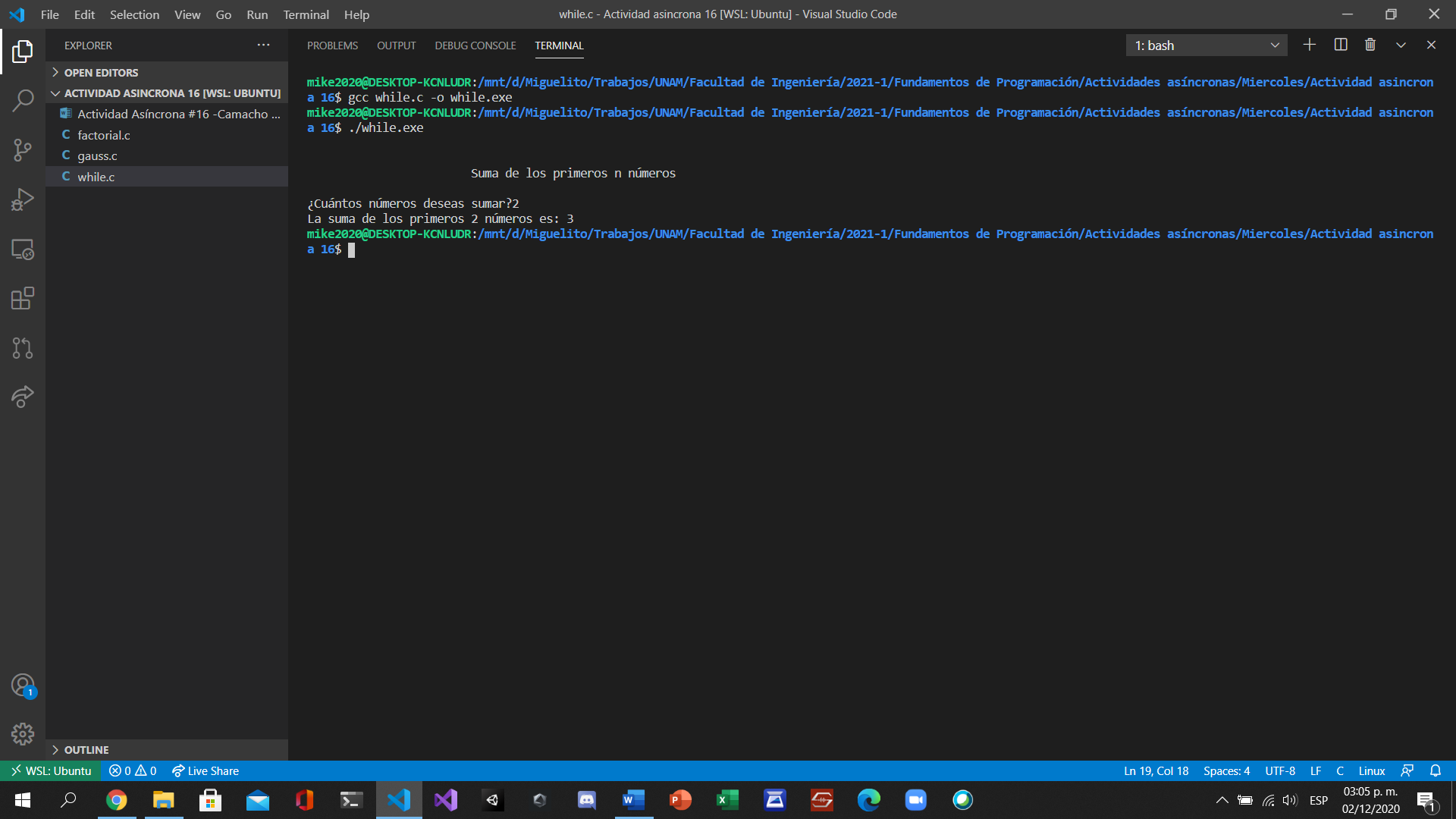
### Ejercicio 4



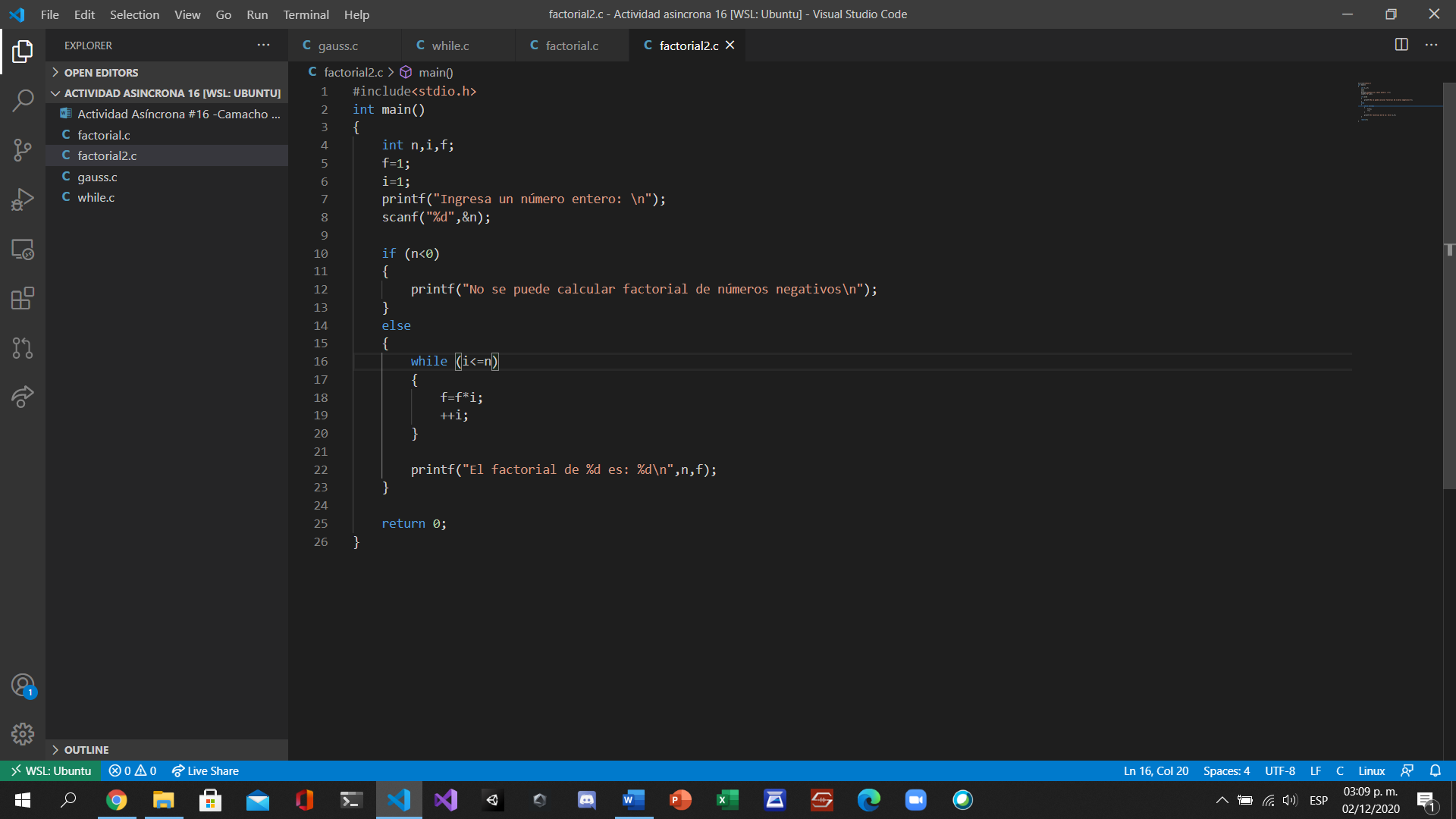


### While.c



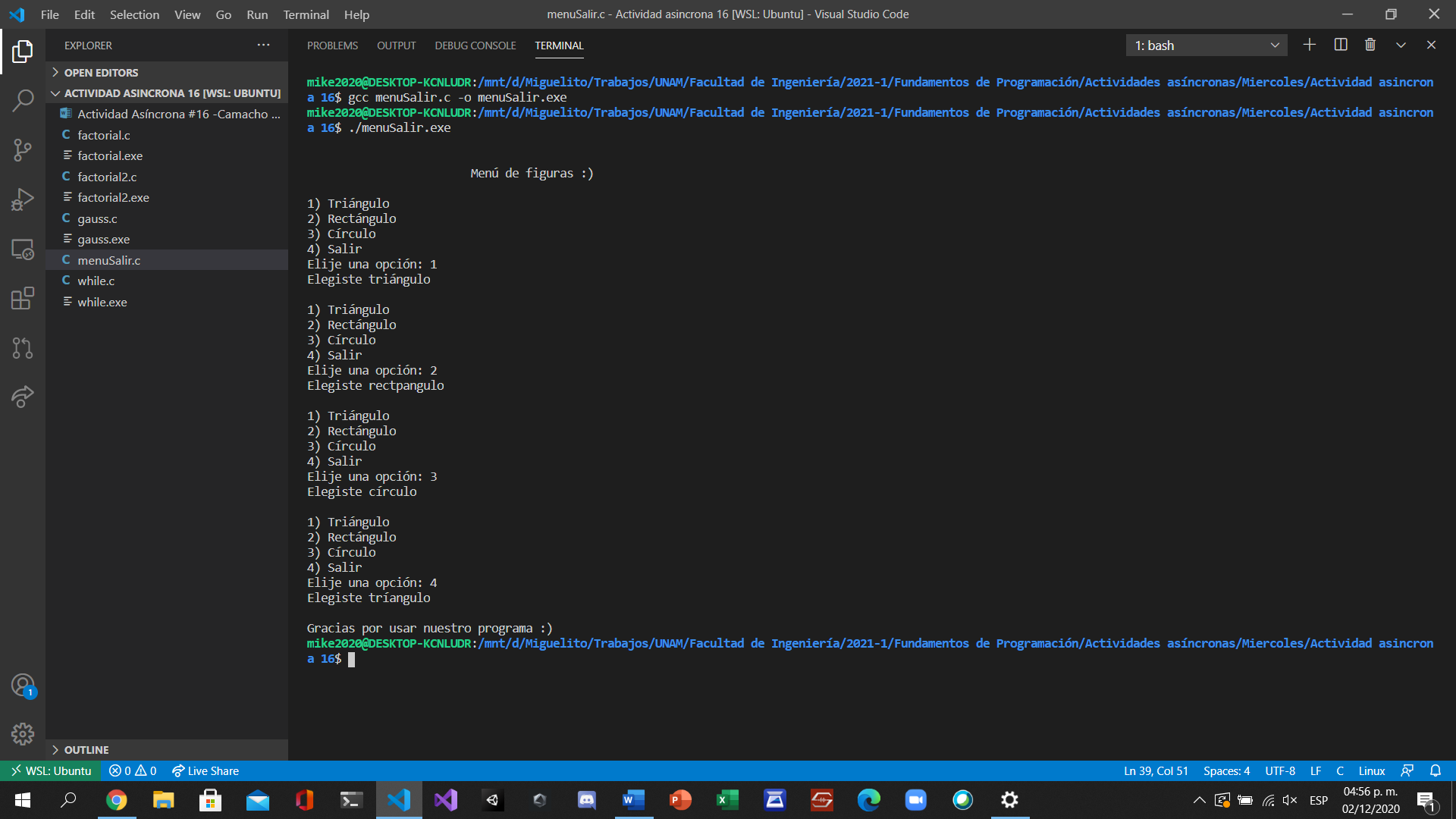
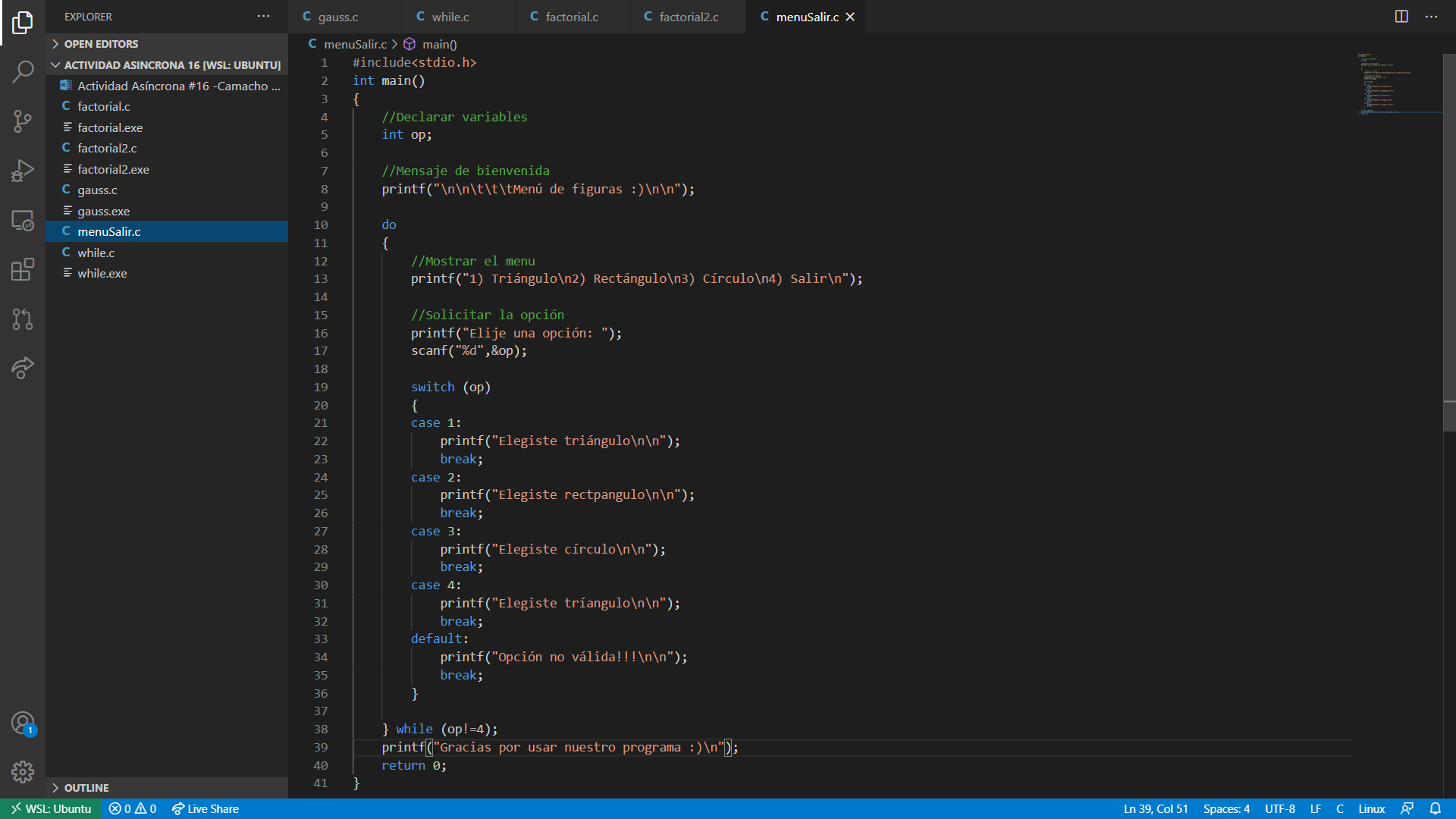


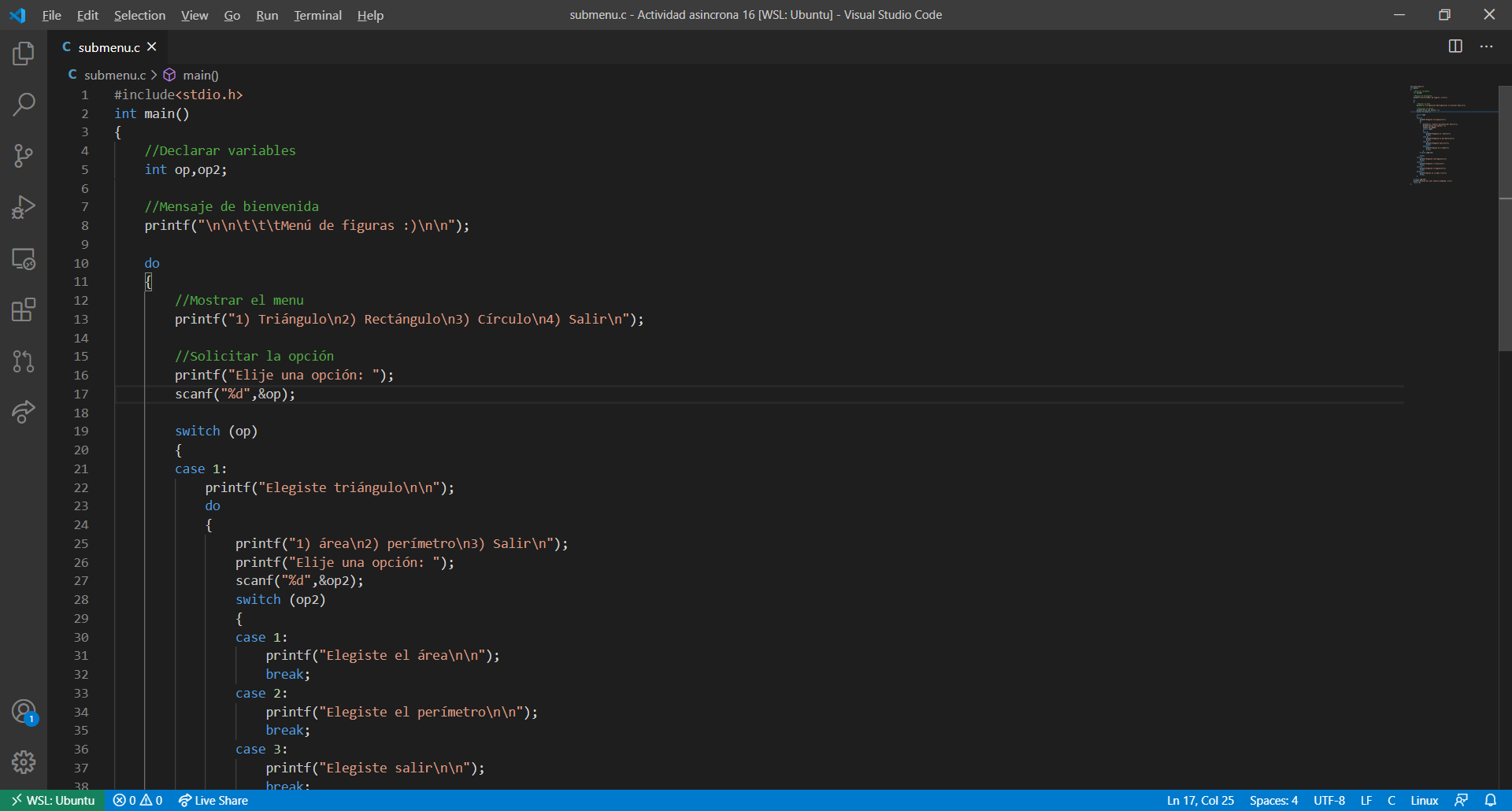
### Factorial2.c



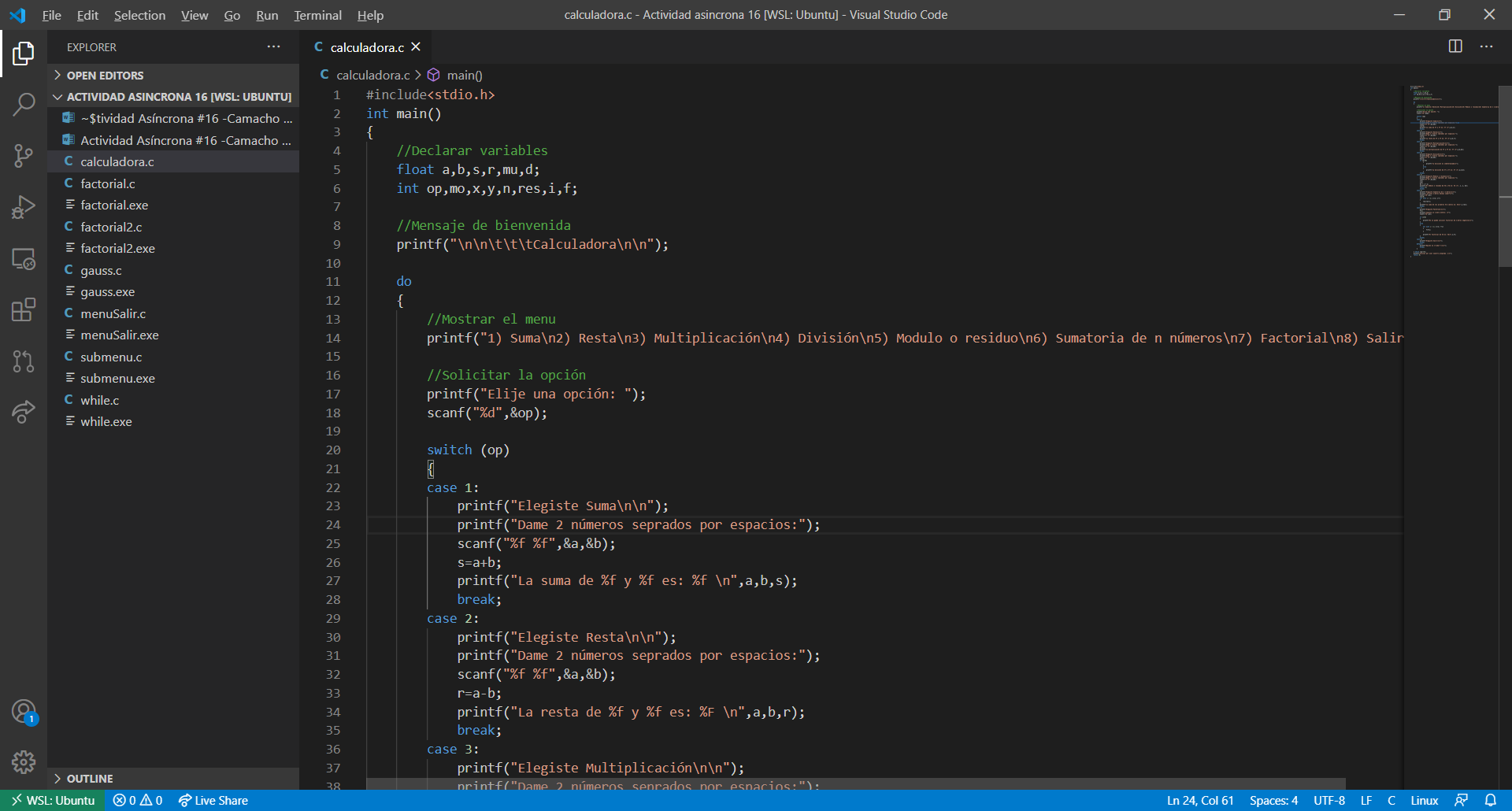


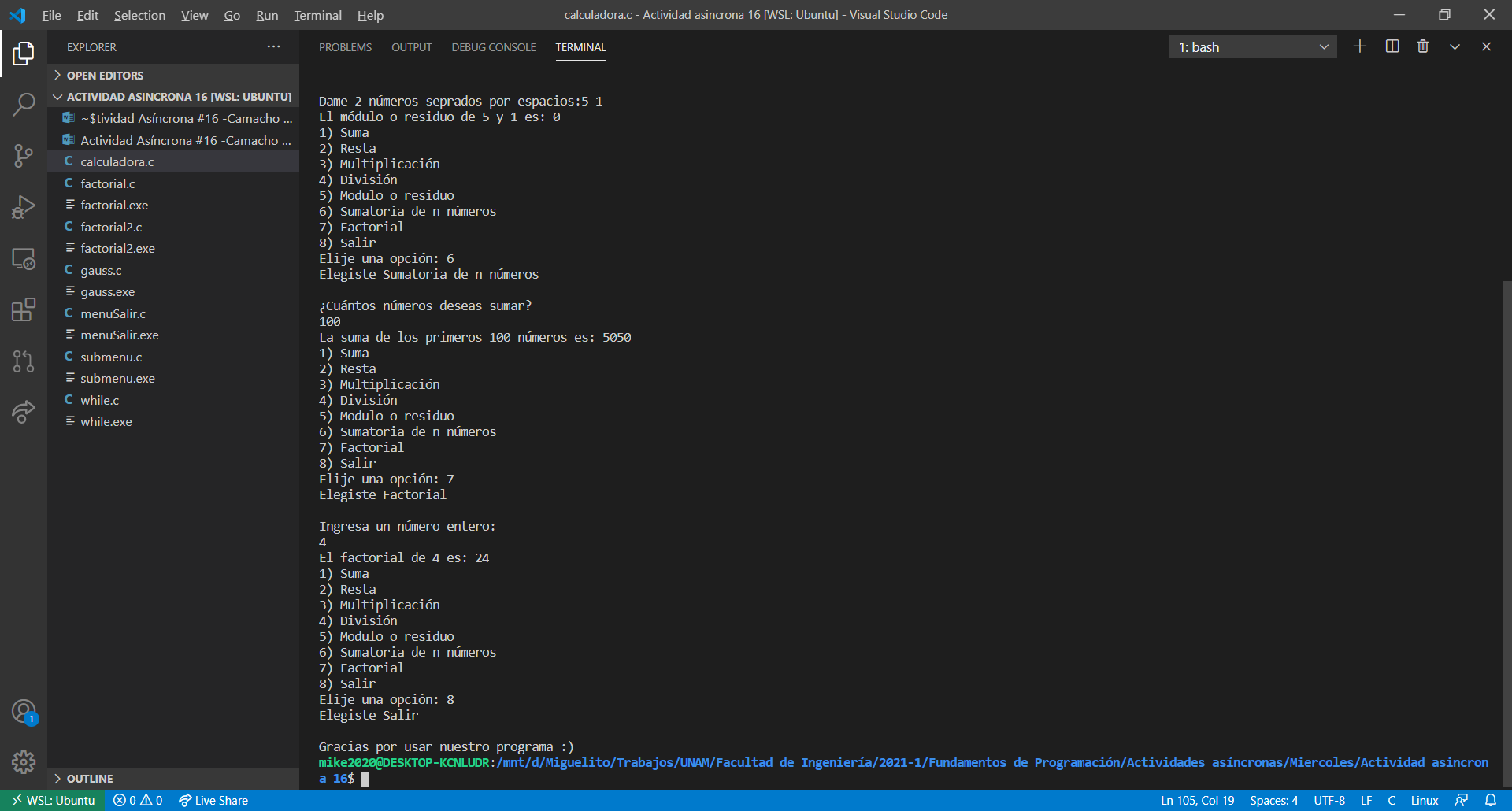
### Menú.c





### Tarea 5





# Conclusiones

Me gustó hacer esta práctica porque puedo identificar las estructuras de control para mi proyecto que es un videojuego usando Unity y C#.

# Referencias

Solano Gálvez, J. A., García Cano, E. E., Sandoval Montaño, L., Nakayama Cervantes, A., Arteaga Ricci, T. I., & Castañeda Perdomo, M. (6 de abril de 2018). *Manual de prácticas de Fundamentos de Programación MADO-17\_FP.* Recuperado el 30 de noviembre de 2020, de Fundamentos de Programación: http://lcp02.fi-b.unam.mx/