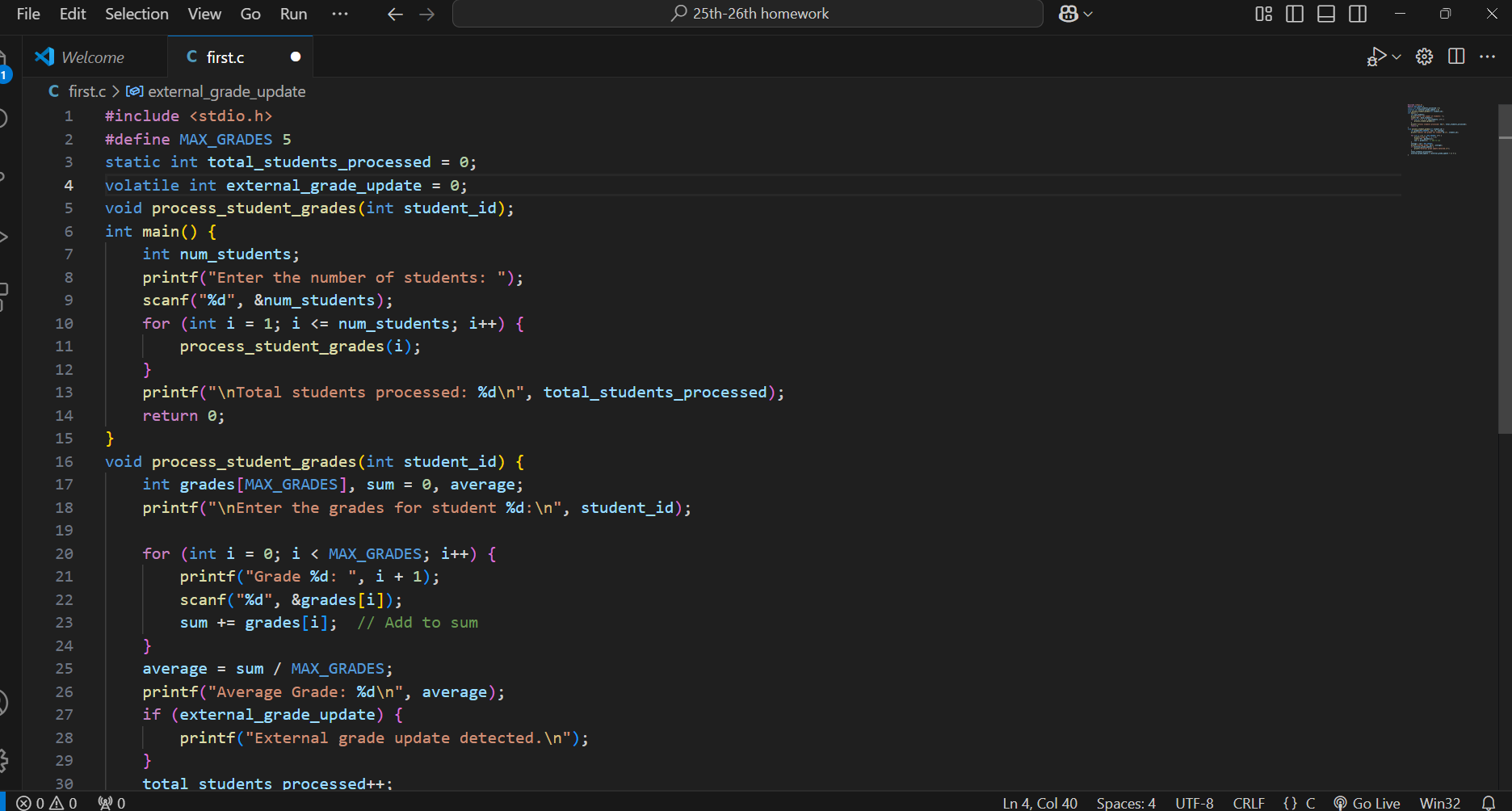
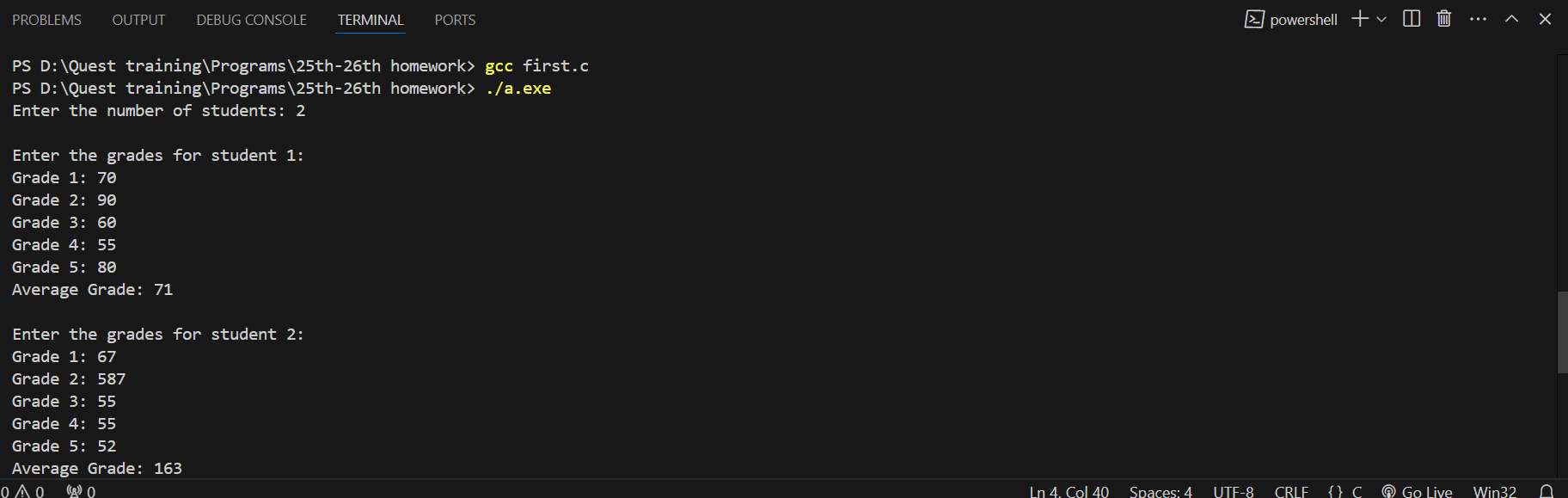
**Homework [25/26-12-2024]**

**1. Student Grade Management System**

**\* Problem Statement: Create a program to manage student grades. Use: A static variable to keep track of the total number of students processed. A const global variable for the maximum number of grades. A volatile variable to simulate an external grade update process. Use if-else and switch to determine grades based on marks and a for loop to process multiple students. Key Concepts Covered: Storage classes (static, volatile), Type qualifiers (const), Decision-making (if-else, switch), Looping (for).**

****

****

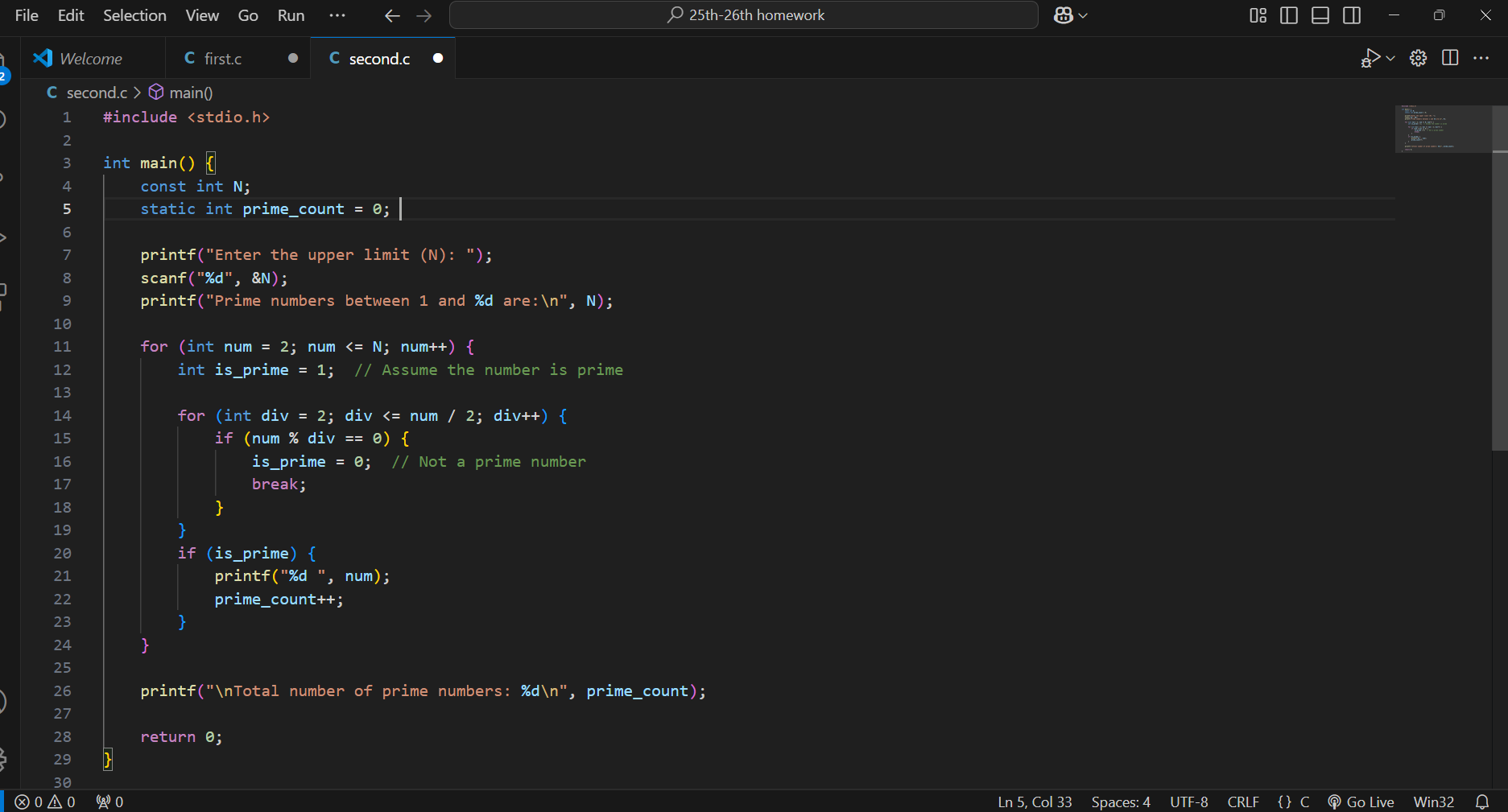
**2. Prime Number Finder**

**Problem Statement: Write a program to find all prime numbers between 1 and a given number N. Use: A const variable for the upper limit N.**

**A static variable to count the total number of prime numbers found.**

**Nested for loops for the prime-checking logic.**

**Key Concepts Covered: Type qualifiers (const), Storage classes (static), Looping (for).**

****

****

**3. Dynamic Menu-Driven Calculator**

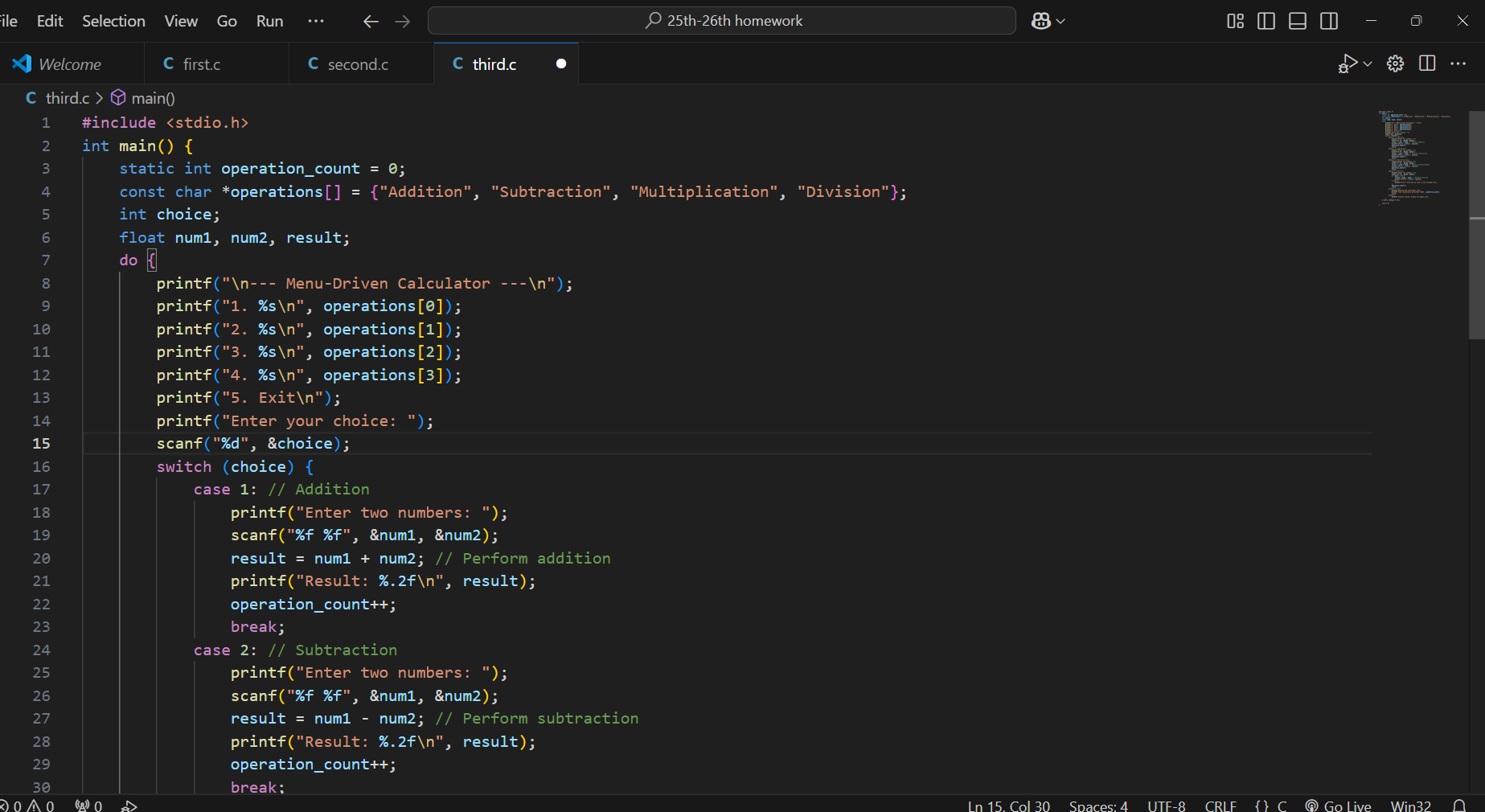
**\* Problem Statement: Create a menu-driven calculator with options for addition, subtraction, multiplication, and division. Use:**

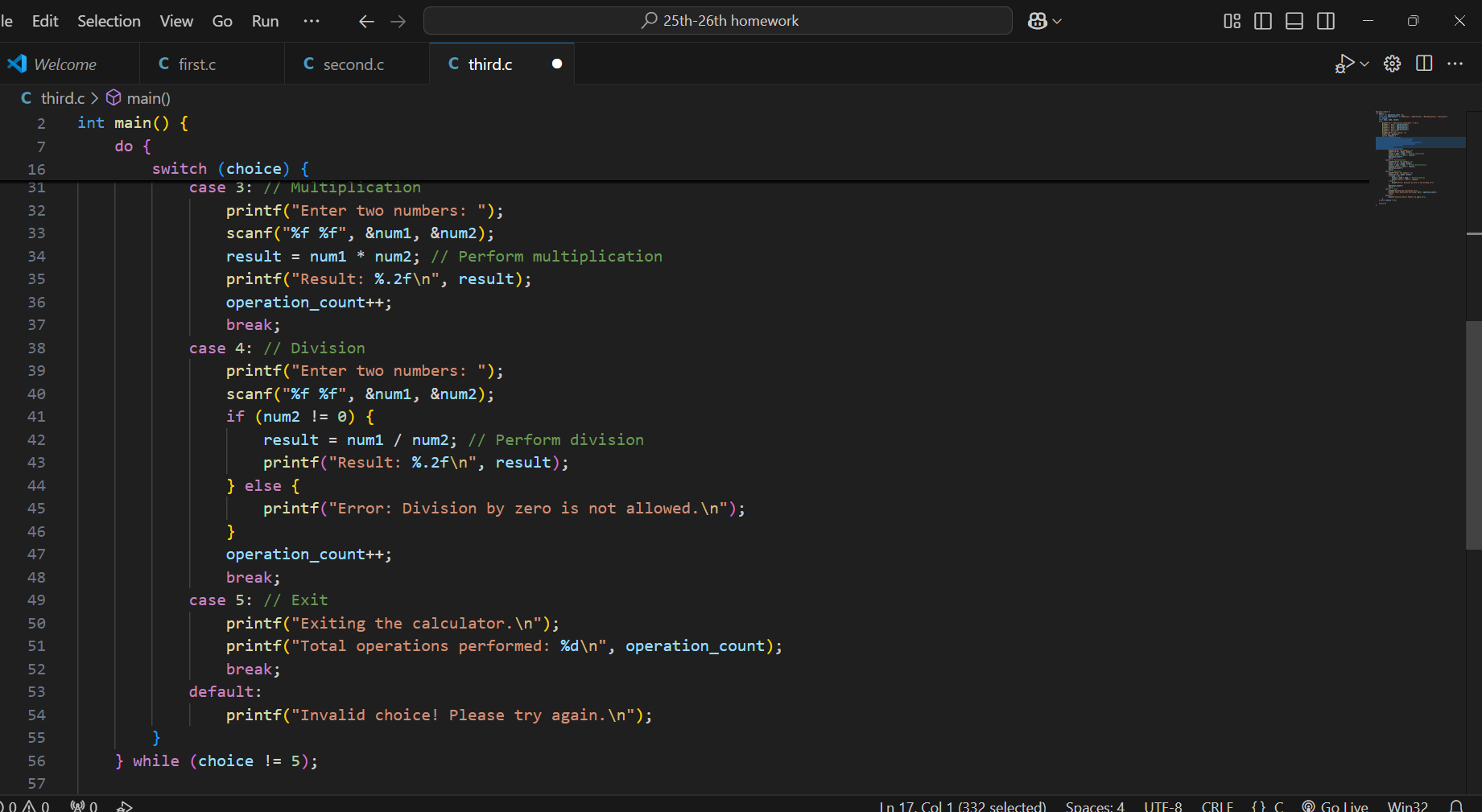
**A static variable to track the total number of operations performed.**

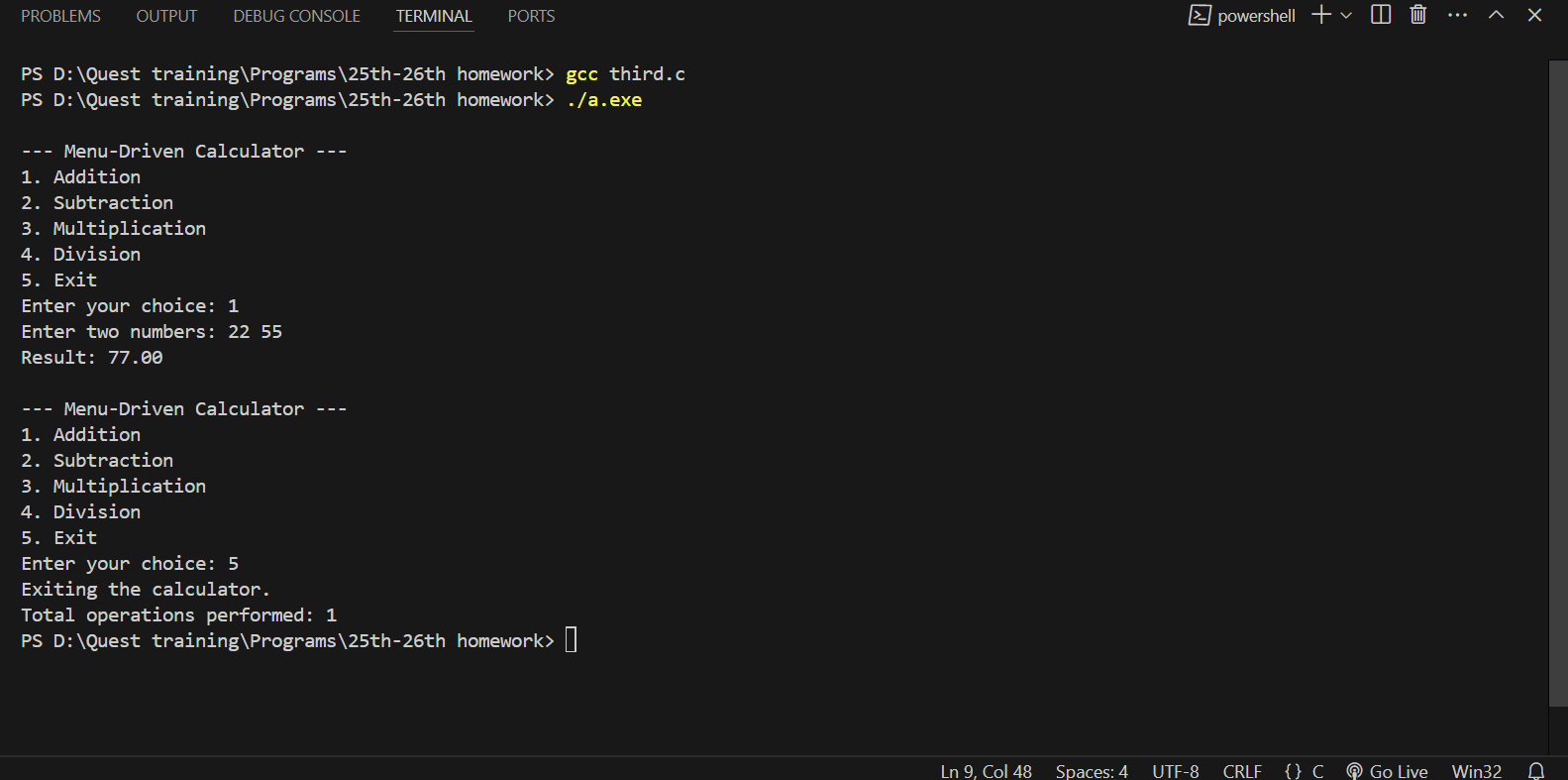
**A const pointer to hold operation names.**

**A do-while loop for the menu and a switch case for operation selection.**

**Key Concepts Covered: Storage classes (static), Type qualifiers (const), Decision-making (switch), Looping (do-while).**

****

****

****

**4. Configuration-Based Matrix Operations**

**\* Problem Statement: Perform matrix addition and multiplication. Use:**

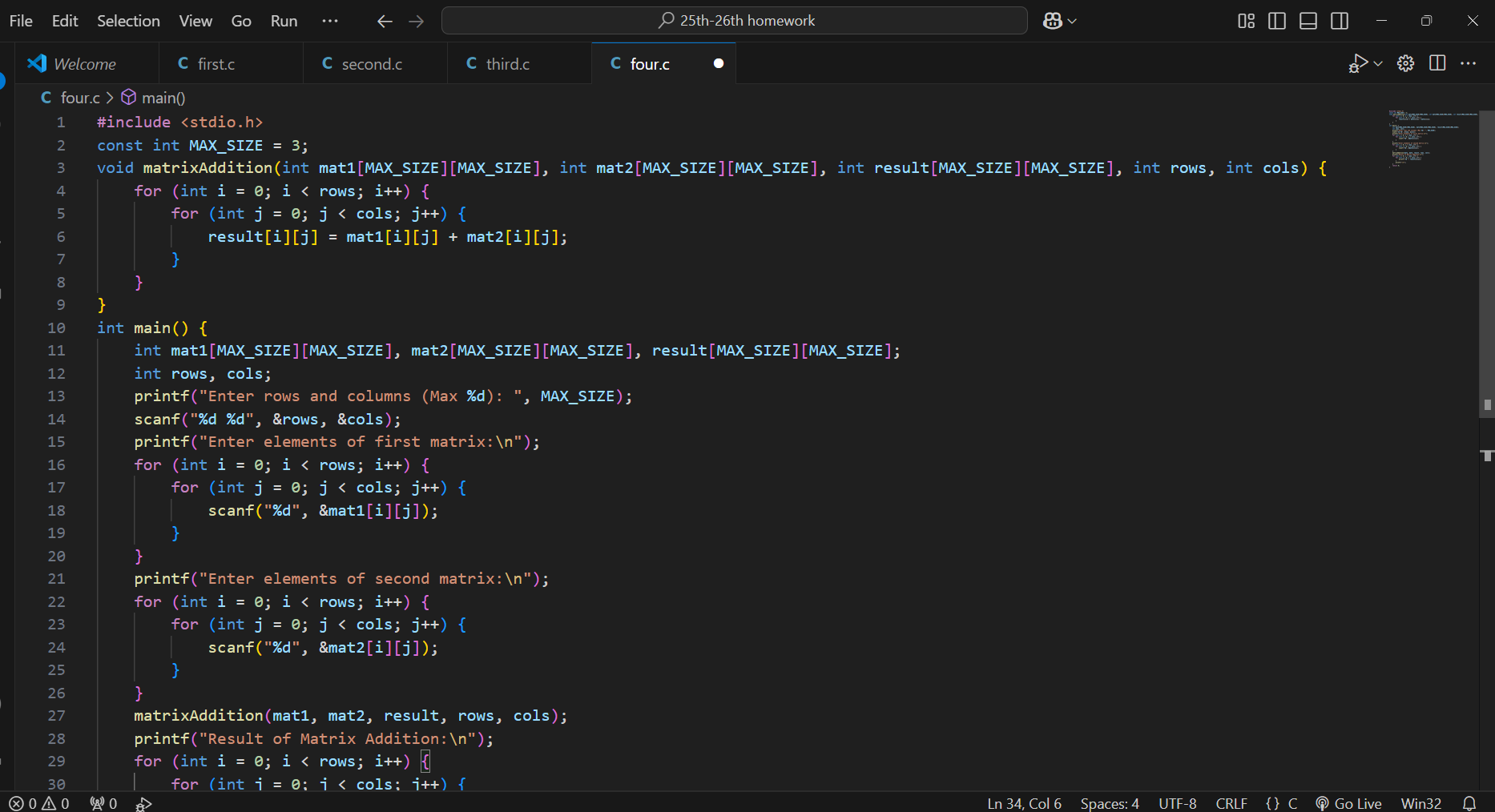
**\* A const global variable to define the maximum size of the matrix.**

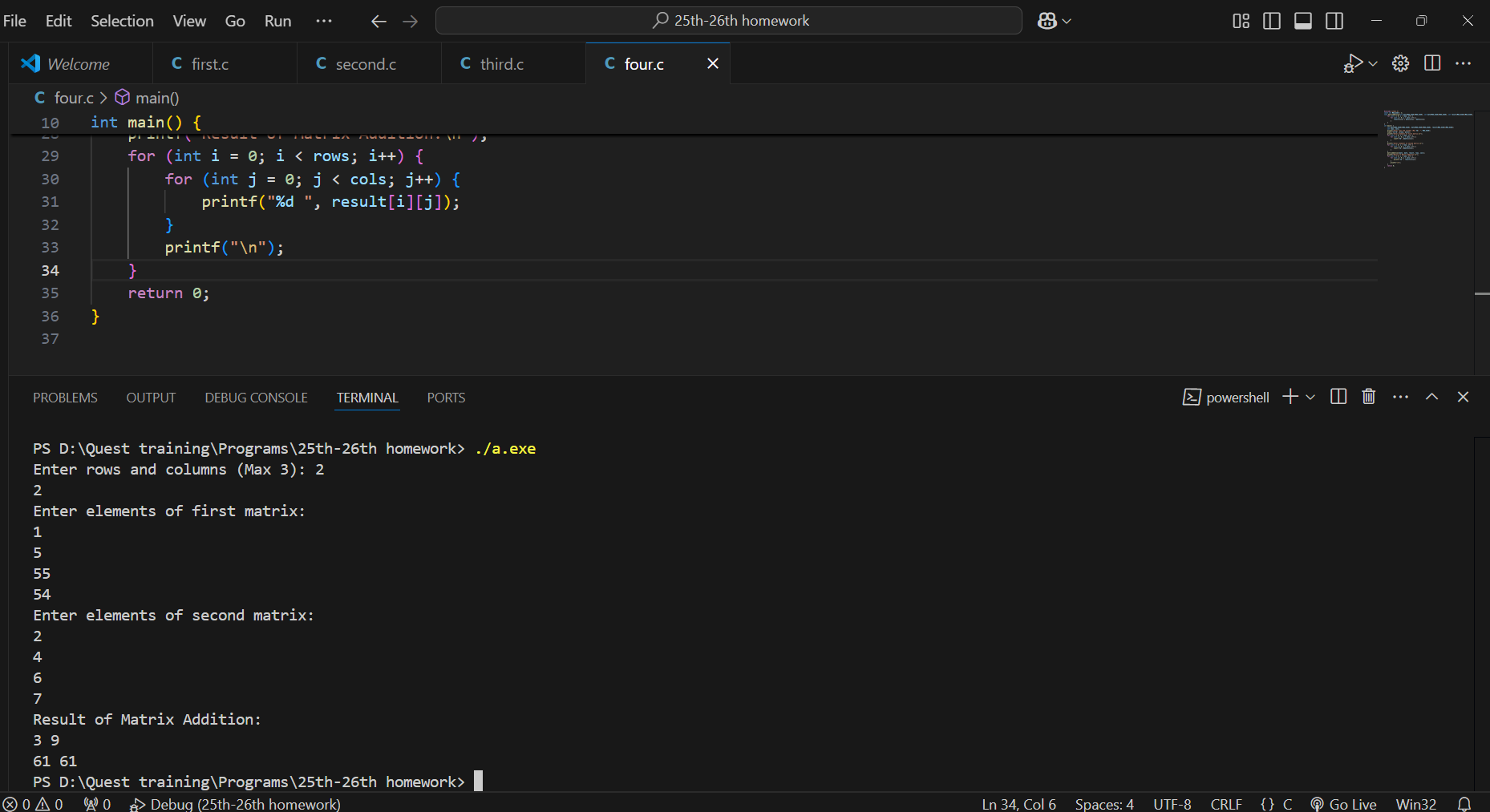
**\* static variables to hold intermediate results.**

**\* if statements to check for matrix compatibility.**

**\* Nested for loops for matrix calculations.**

**\* Key Concepts Covered: Type qualifiers (const), Storage classes (static), Decision-making (if), Looping (nested for).**

****

****

**5. Temperature Monitoring System**

**\* Problem Statement: Simulate a temperature monitoring system using:**

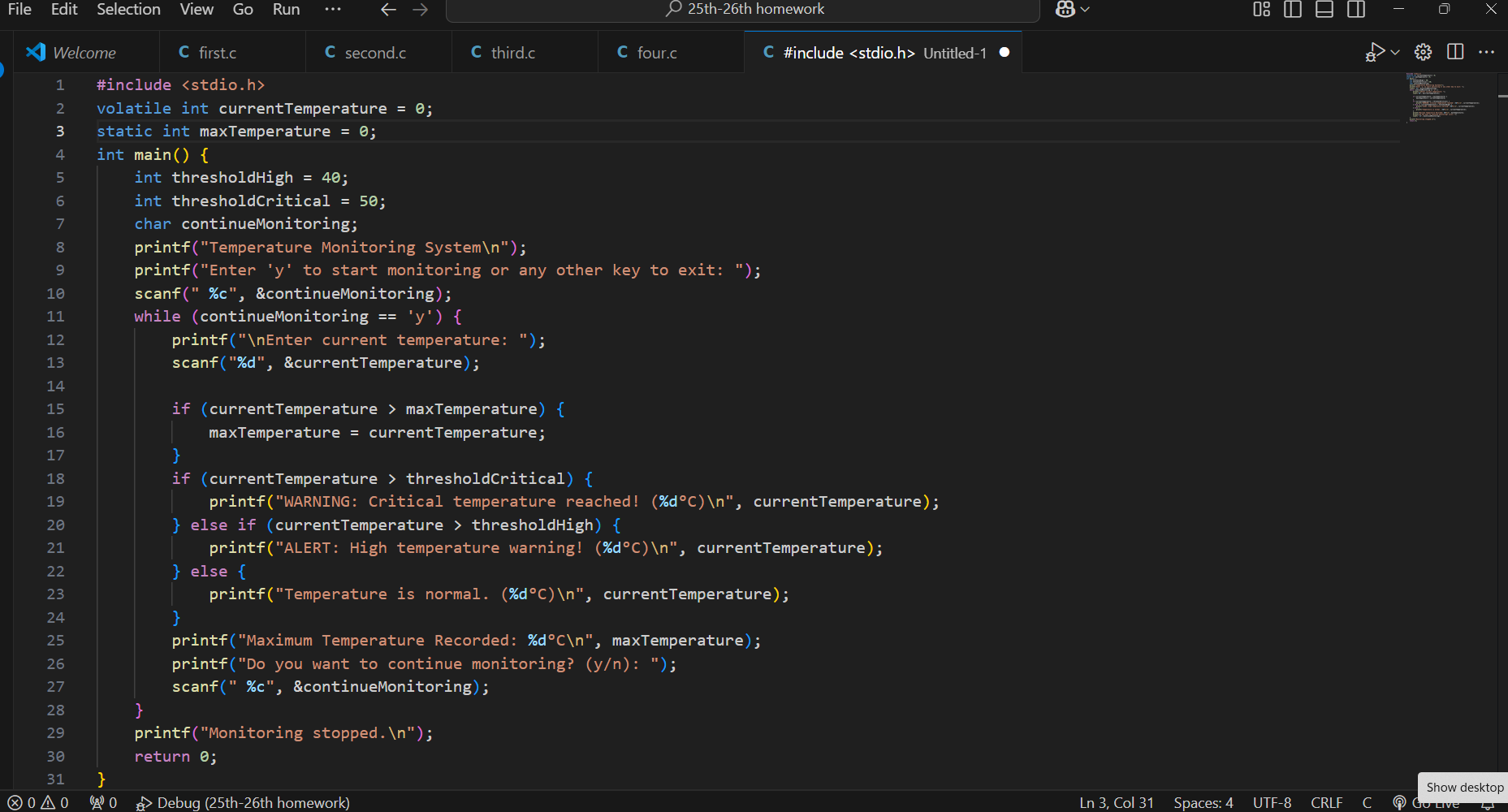
**\* A volatile variable to simulate temperature input.**

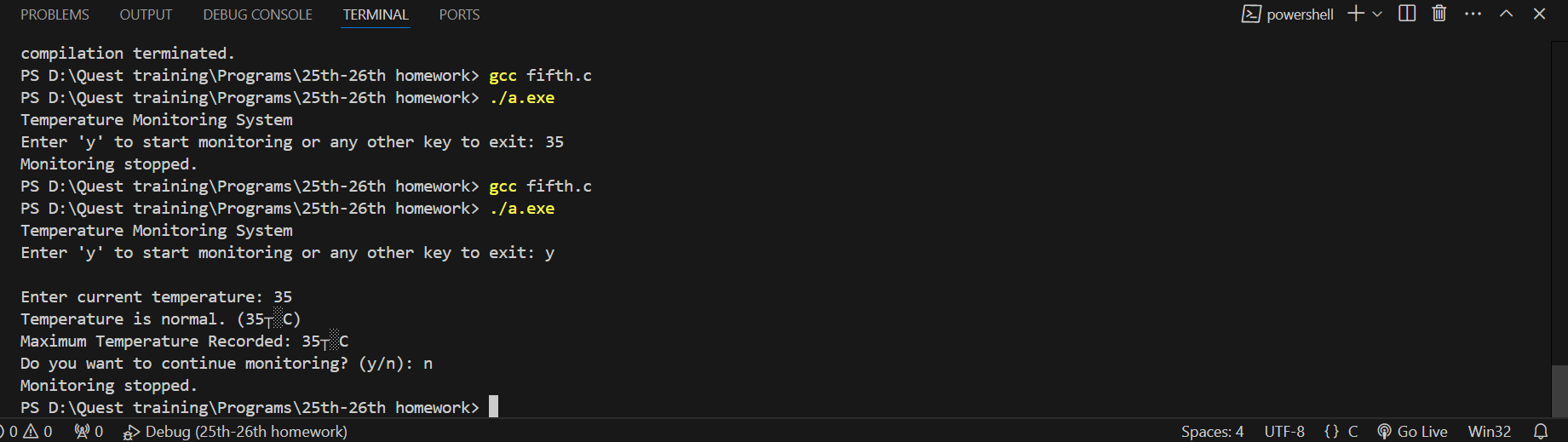
**\* A static variable to hold the maximum temperature recorded.**

**\* if-else statements to issue warnings when the temperature exceeds thresholds.**

**\* A while loop to continuously monitor and update the temperature.**

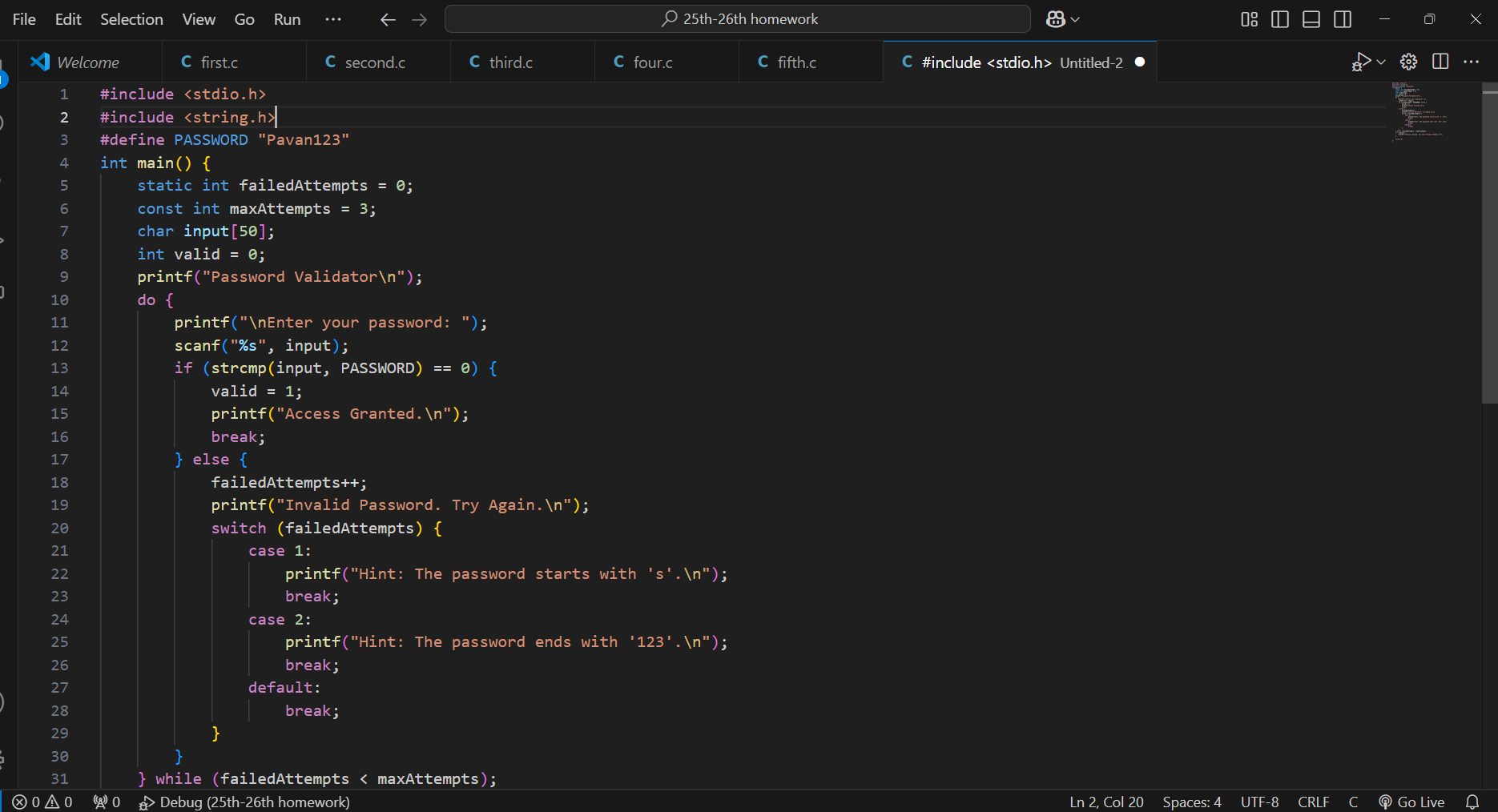
**\* Key Concepts Covered: Storage classes (volatile, static), Decision-making (if-else), Looping (while).**

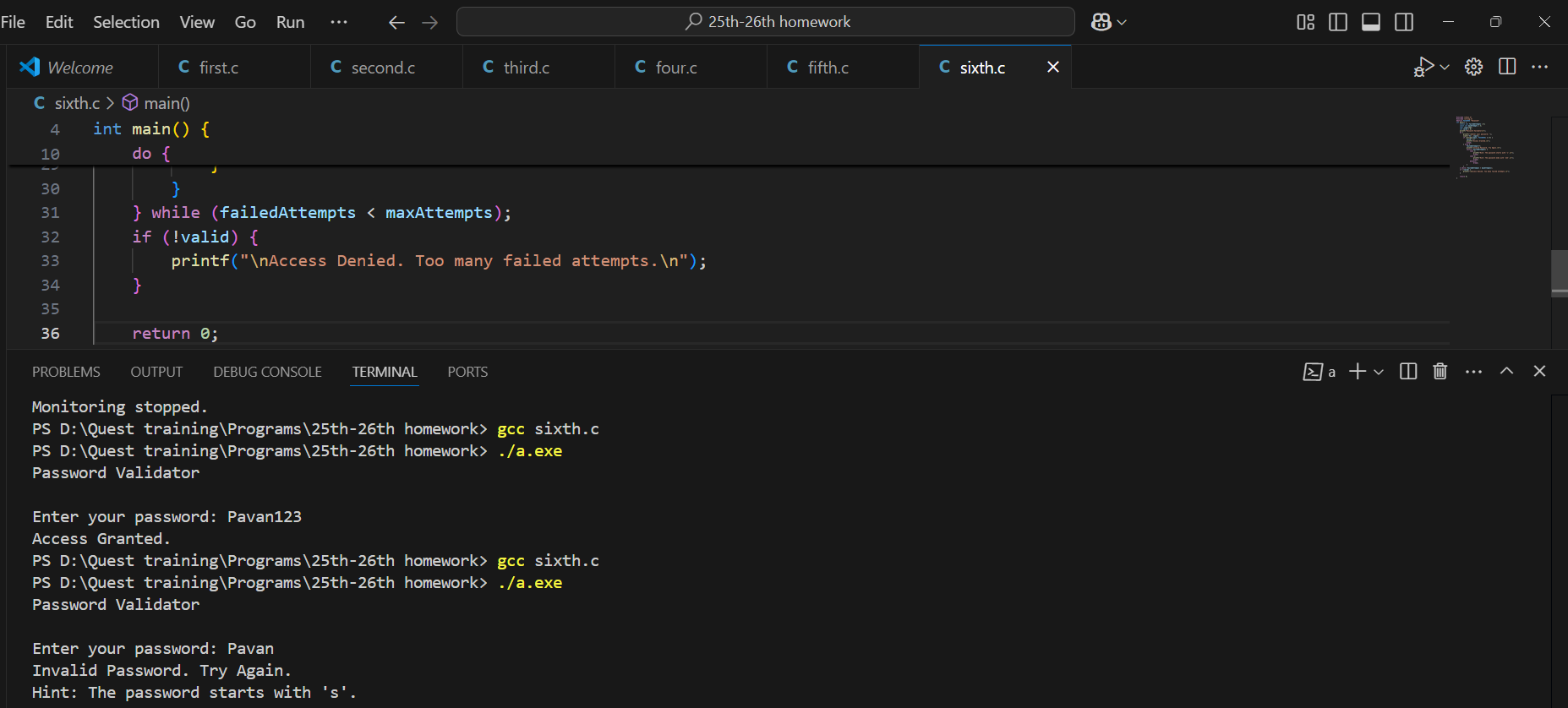
****

****

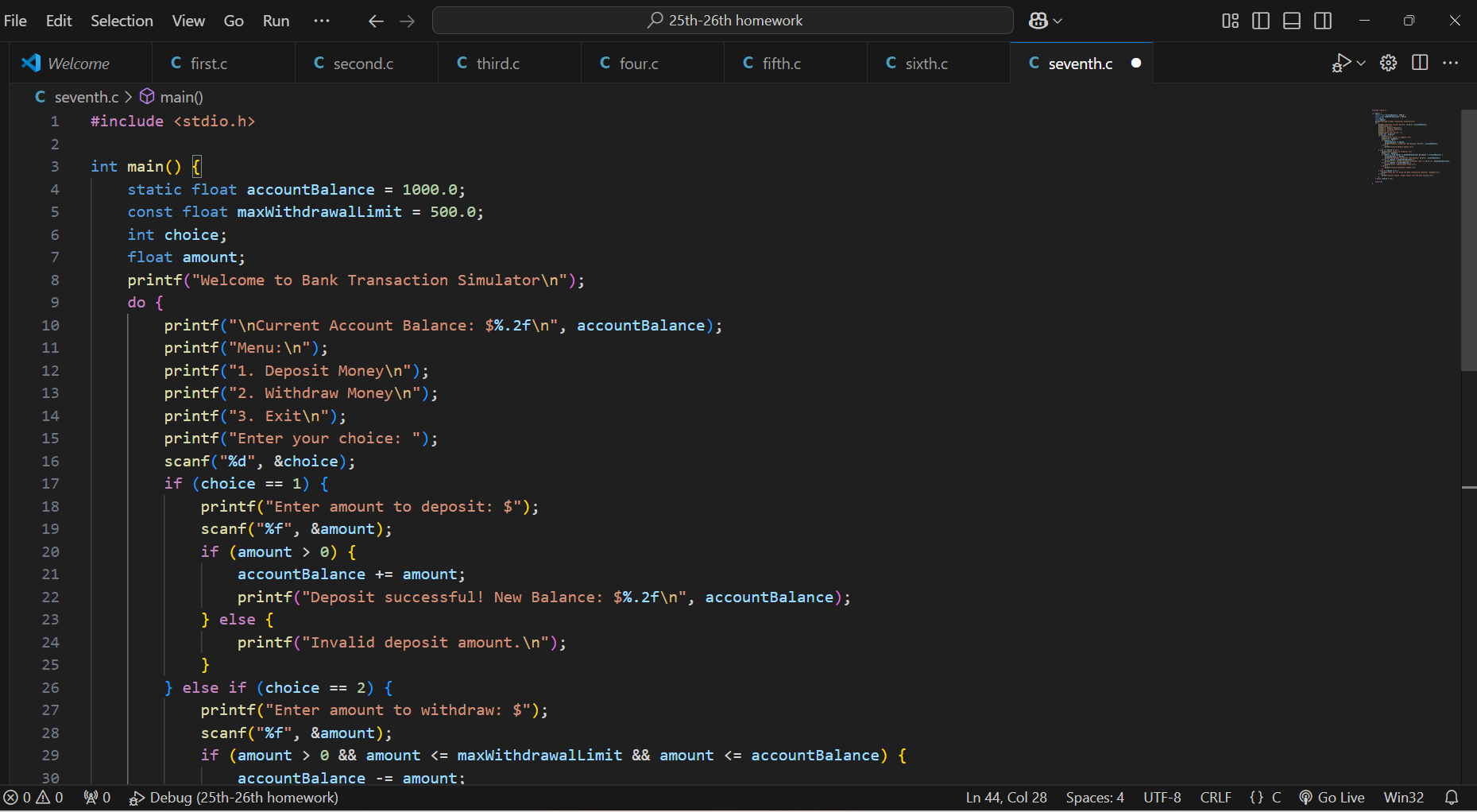
**6. Password Validator - Problem Statement: Implement a password validation program. Use:**

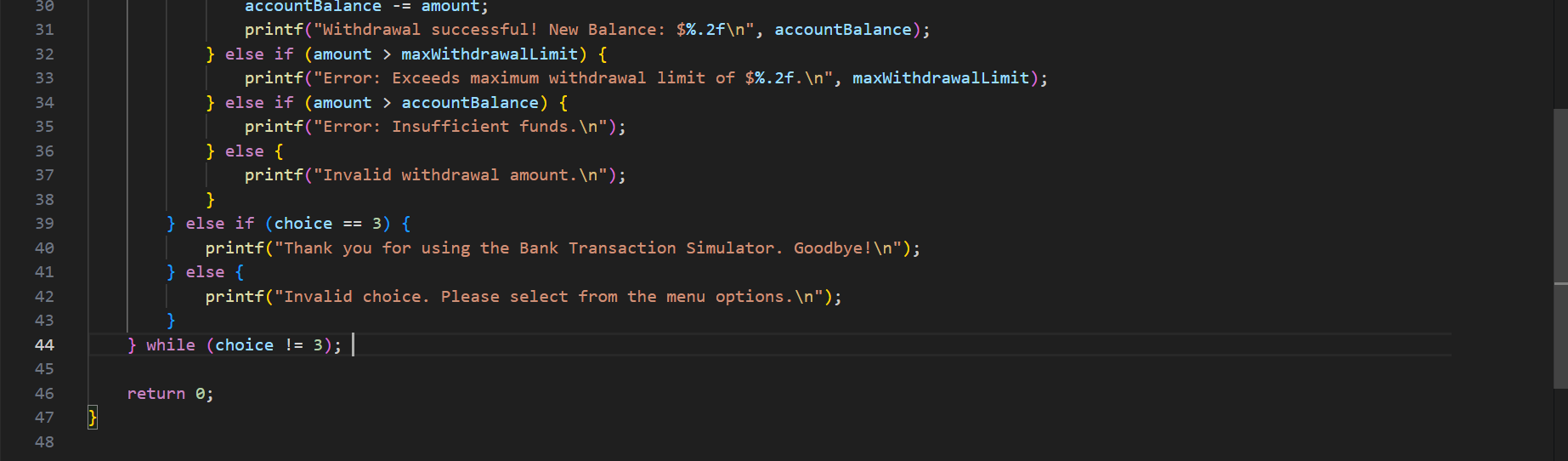
**\* A static variable to count the number of failed attempts. A const variable for the maximum allowed attempts. if-else and switch statements to handle validation rules. A do-while loop to retry password entry. Key Concepts Covered: Storage classes (static), Type qualifiers (const), Decision-making (if-else, switch), Looping (do-while).**

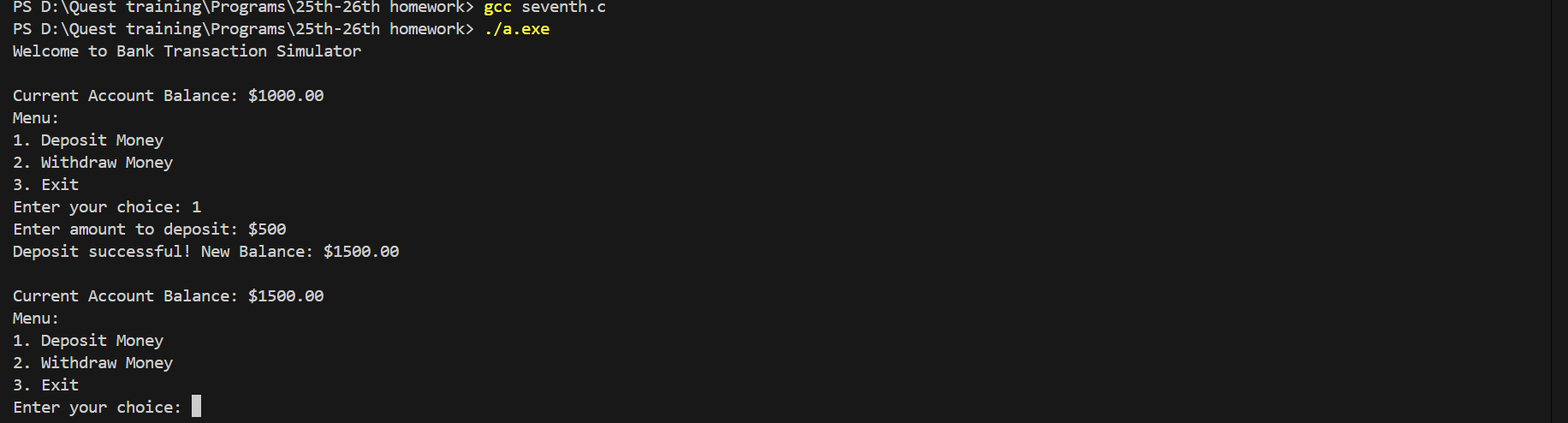
****

****

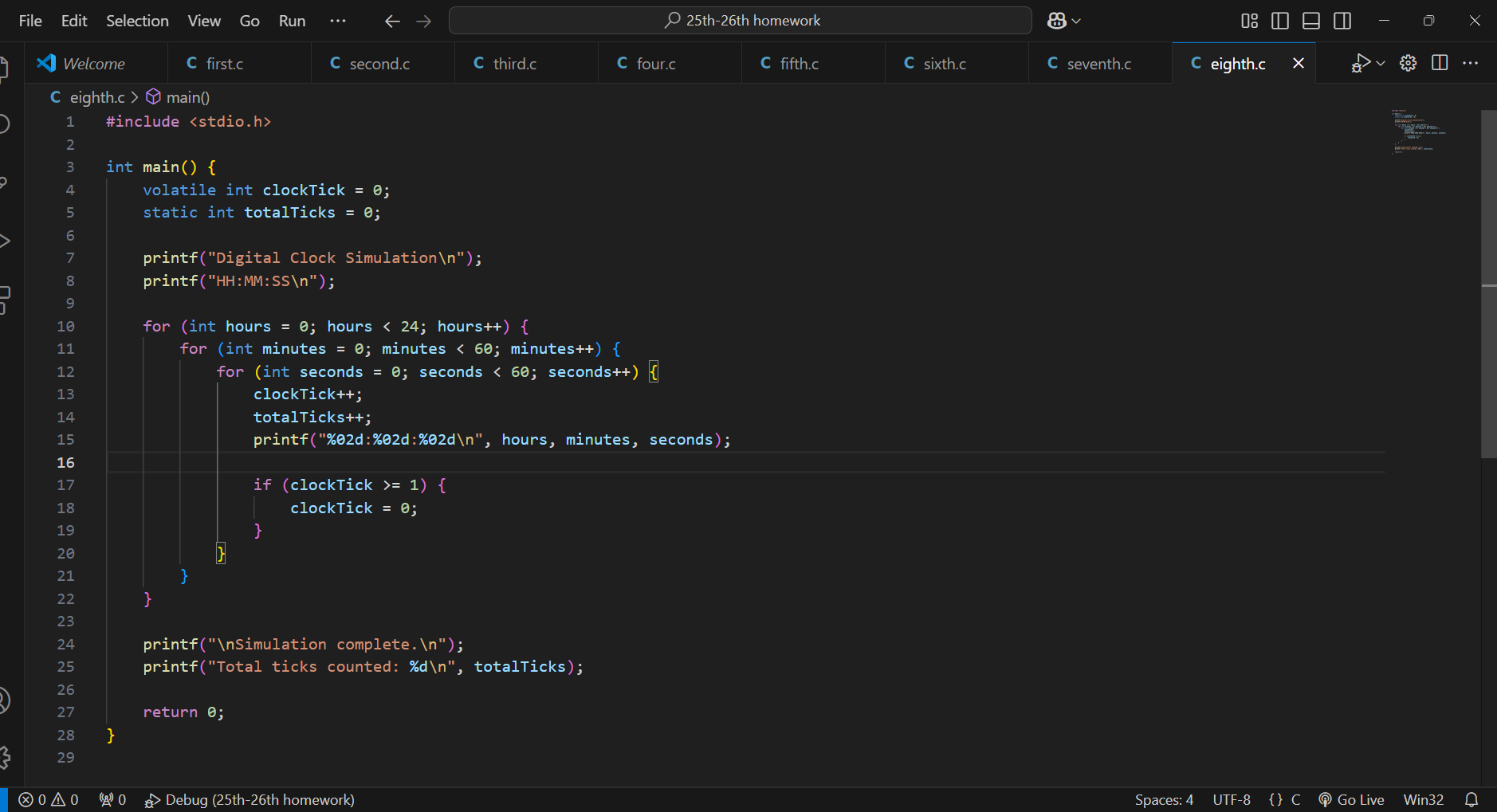
**7. Bank Transaction Simulator - Problem Statement: Simulate bank transactions. Use: A static variable to maintain the account balance.A const variable for the maximum withdrawal limit.if-else statements to check transaction validity.A do-while loop for performing multiple transactions. Key Concepts Covered: Storage classes (static), Type qualifiers (const), Decision-making (if-else), Looping (do-while).**

****

****

****

**8. Digital Clock Simulation - Problem Statement: Simulate a digital clock. Use: volatile variables to simulate clock ticks. A static variable to count the total number of ticks.Nested for loops for hours, minutes, and seconds.if statements to reset counters at appropriate limits. Key Concepts Covered: Storage classes (volatile, static), Decision-making (if), Looping (nested for).**

****

****

**9. Game Score Tracker**

**\* Problem Statement: Track scores in a simple game. Use:**

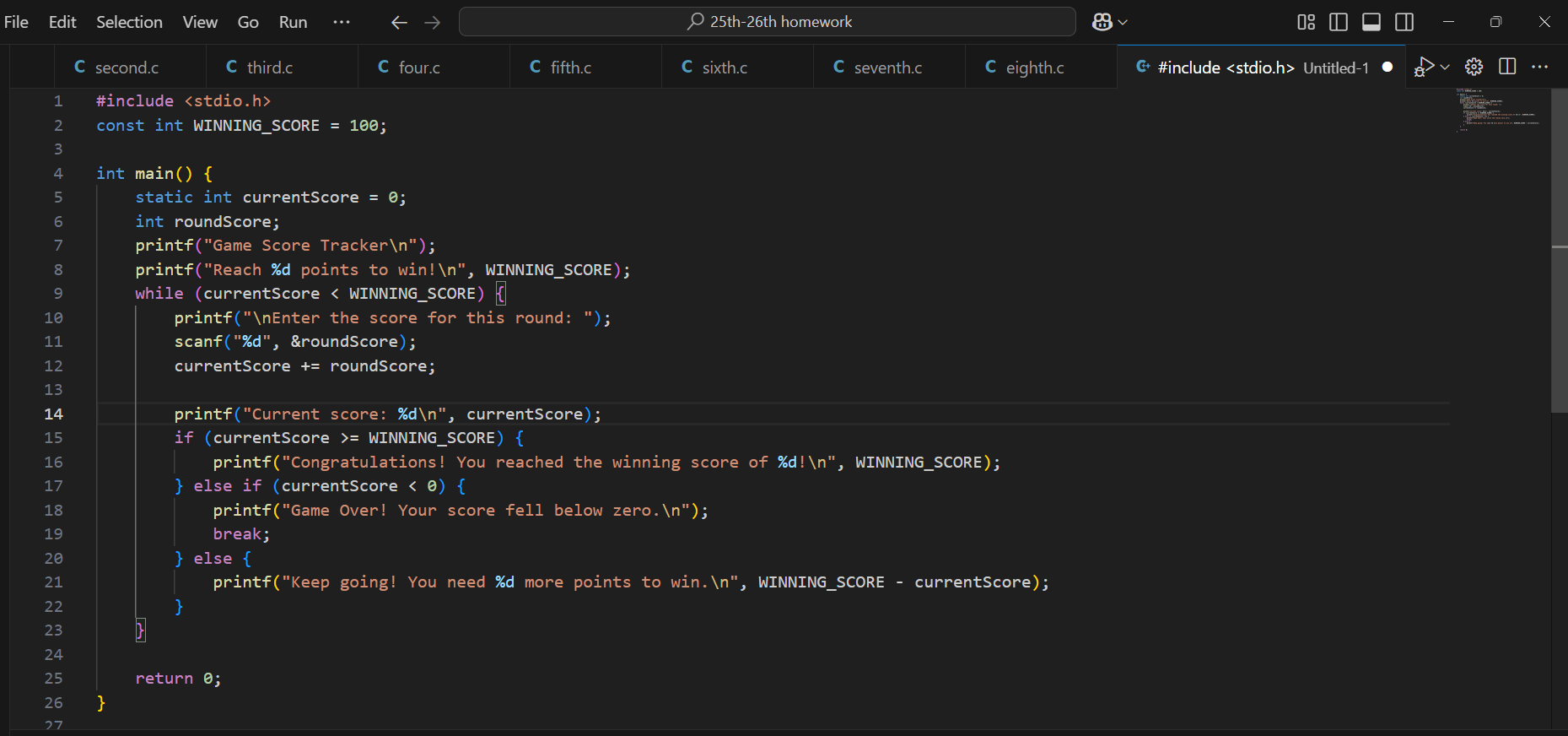
**\* A static variable to maintain the current score.**

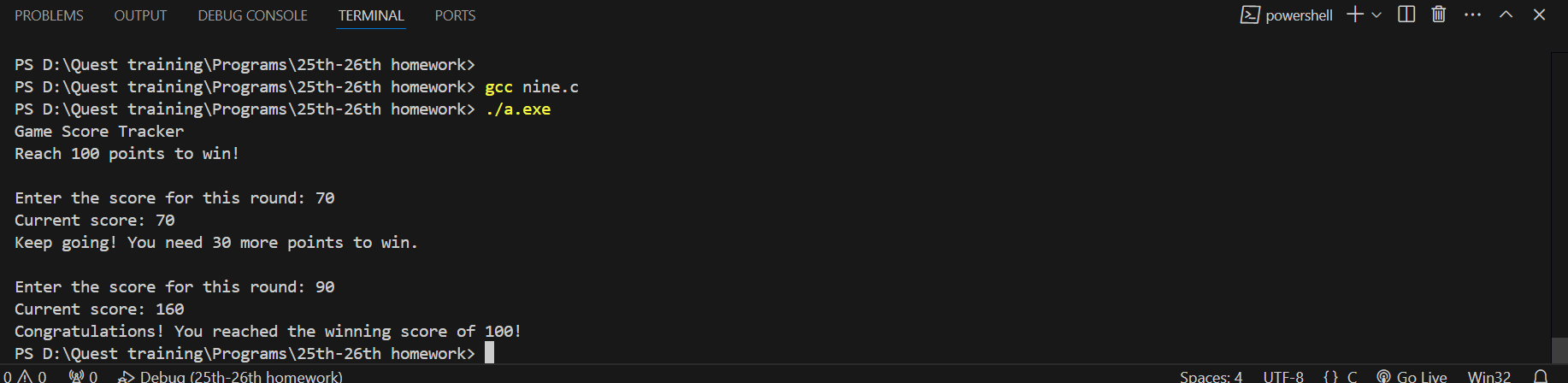
**\* A const variable for the winning score.**

**\* if-else statements to decide if the player has won or lost.**

**\* A while loop to play rounds of the game.**

**\* Key Concepts Covered: Storage classes (static), Type qualifiers (const), Decision-making (if-else), Looping (while).**

****

****