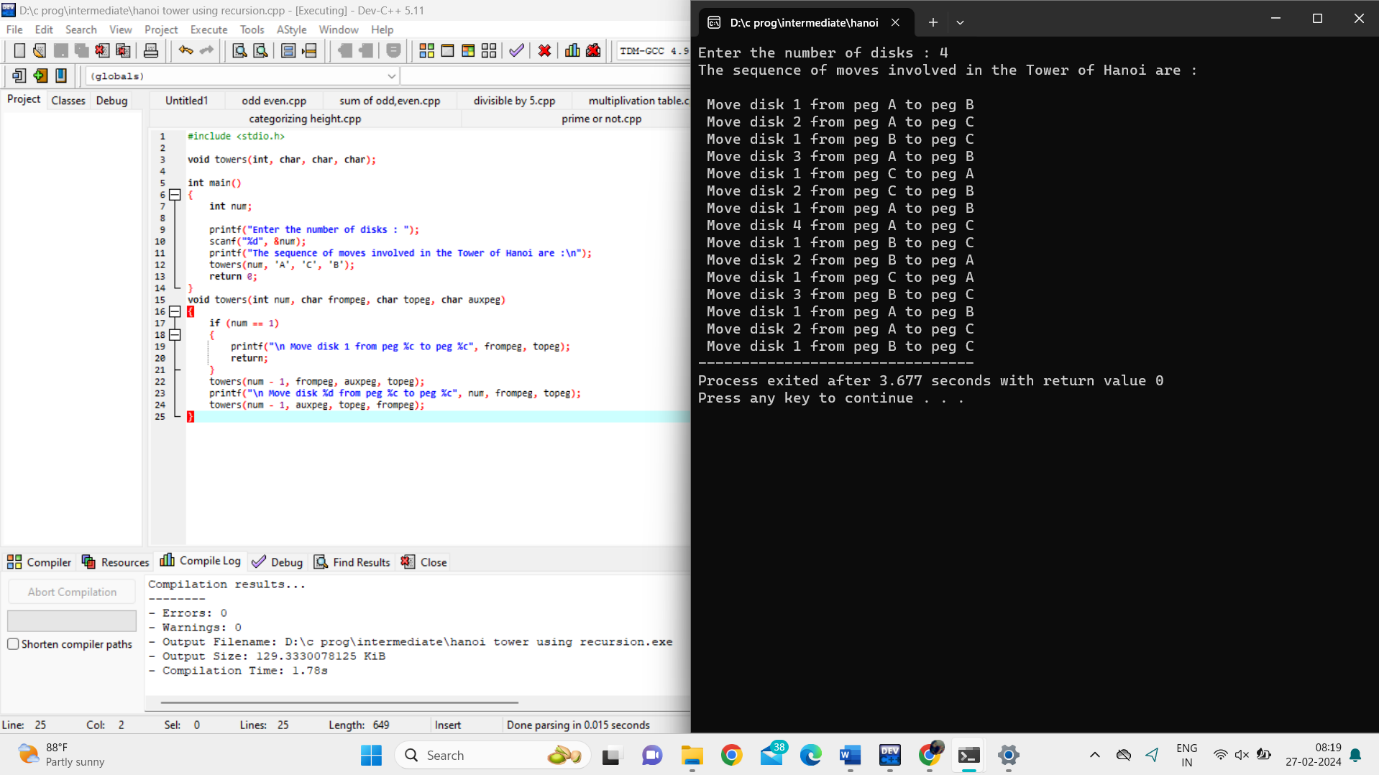
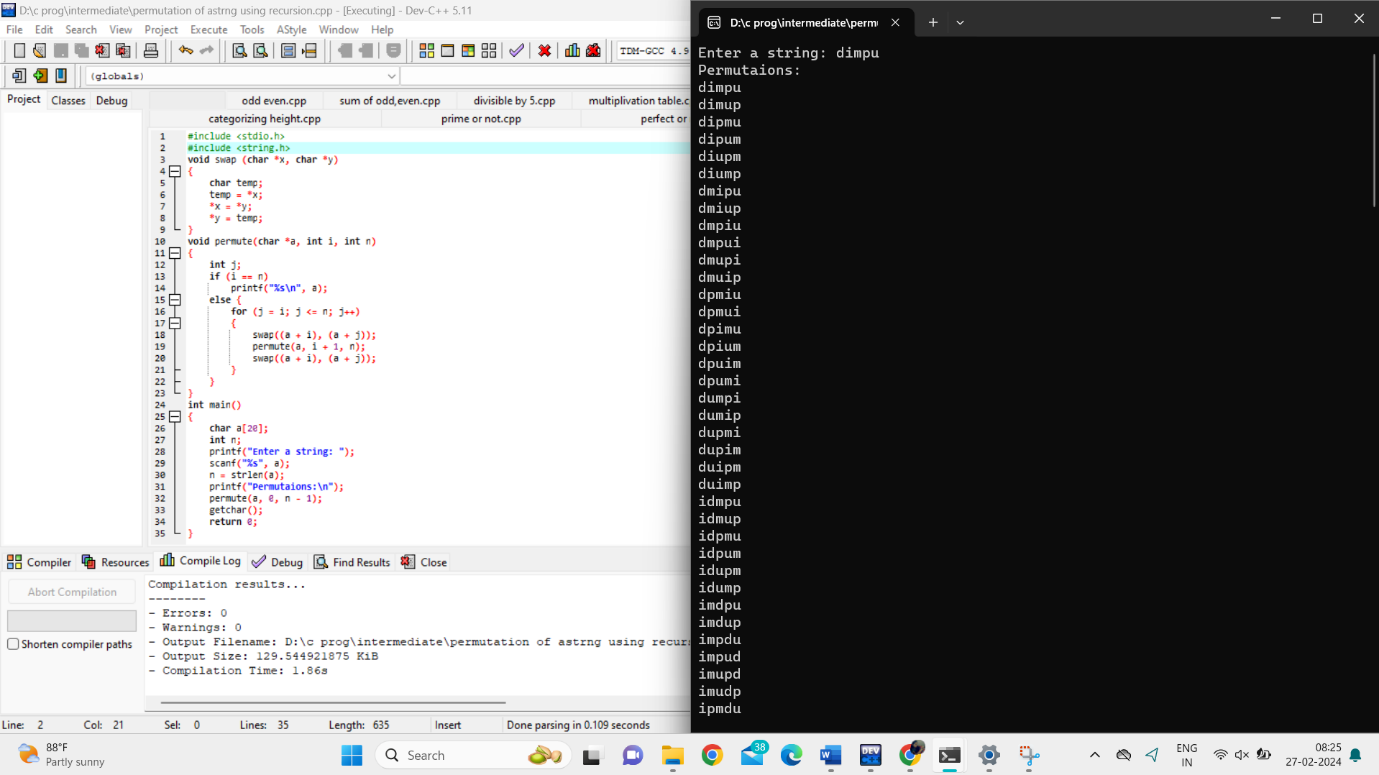
Day 6-c programming

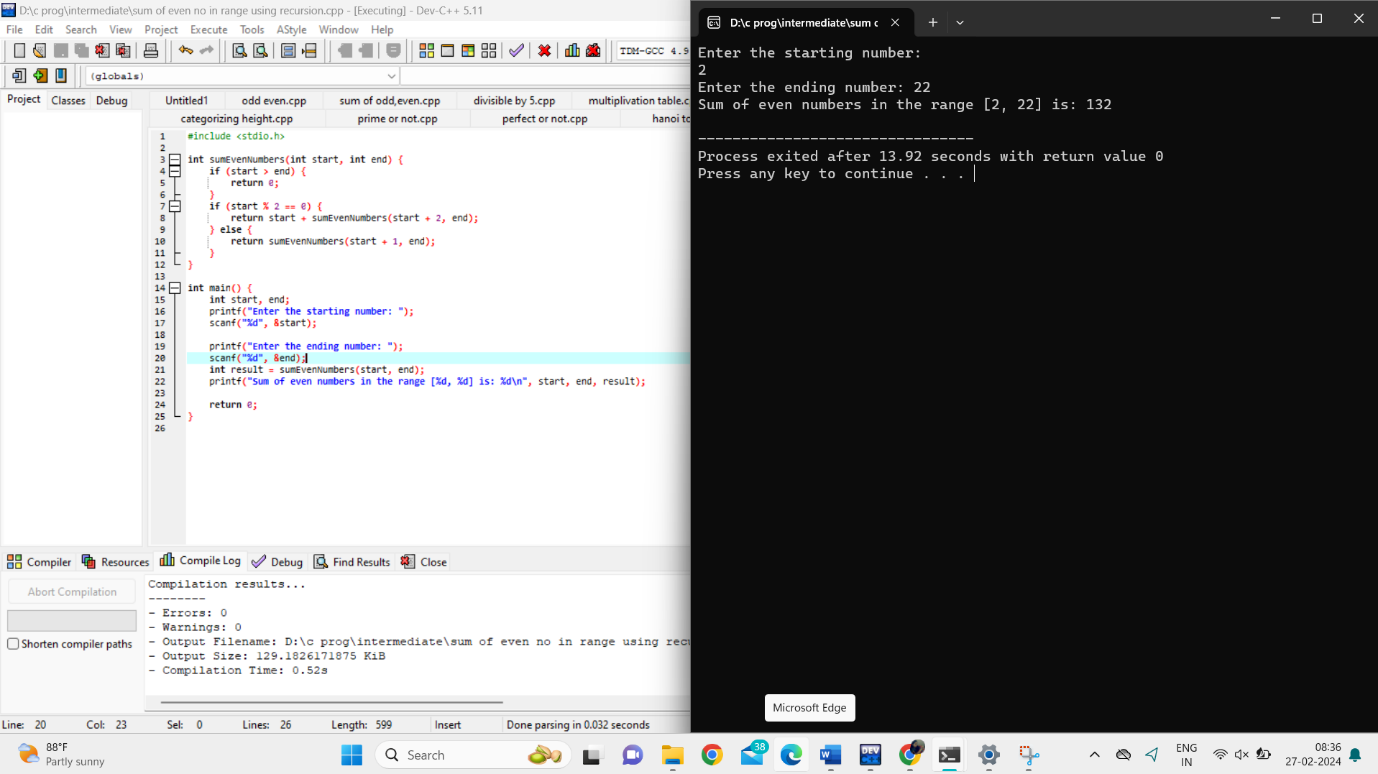
1. Write a program to recursively solve the Tower of Hanoi problem for n disks.



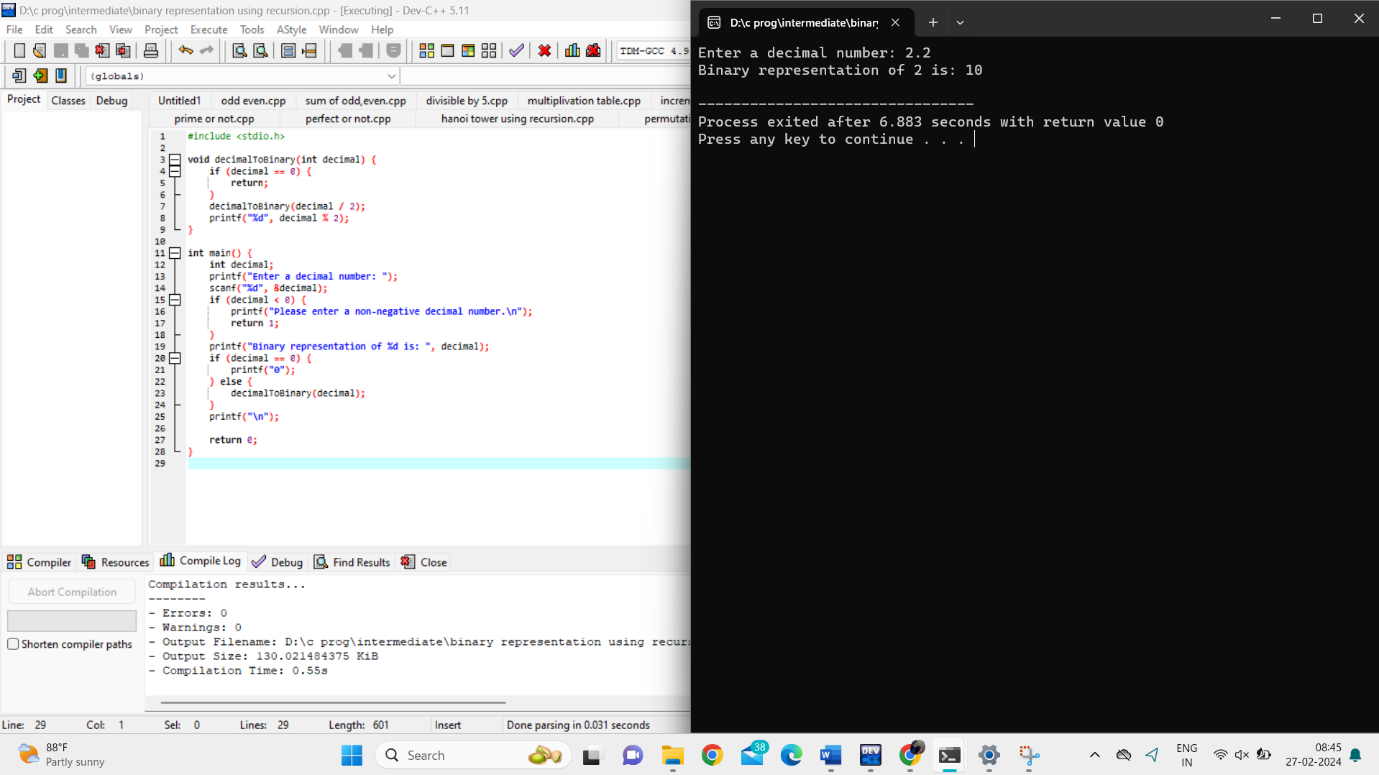
1. Implement a recursive program to generate all possible permutations of a given string.



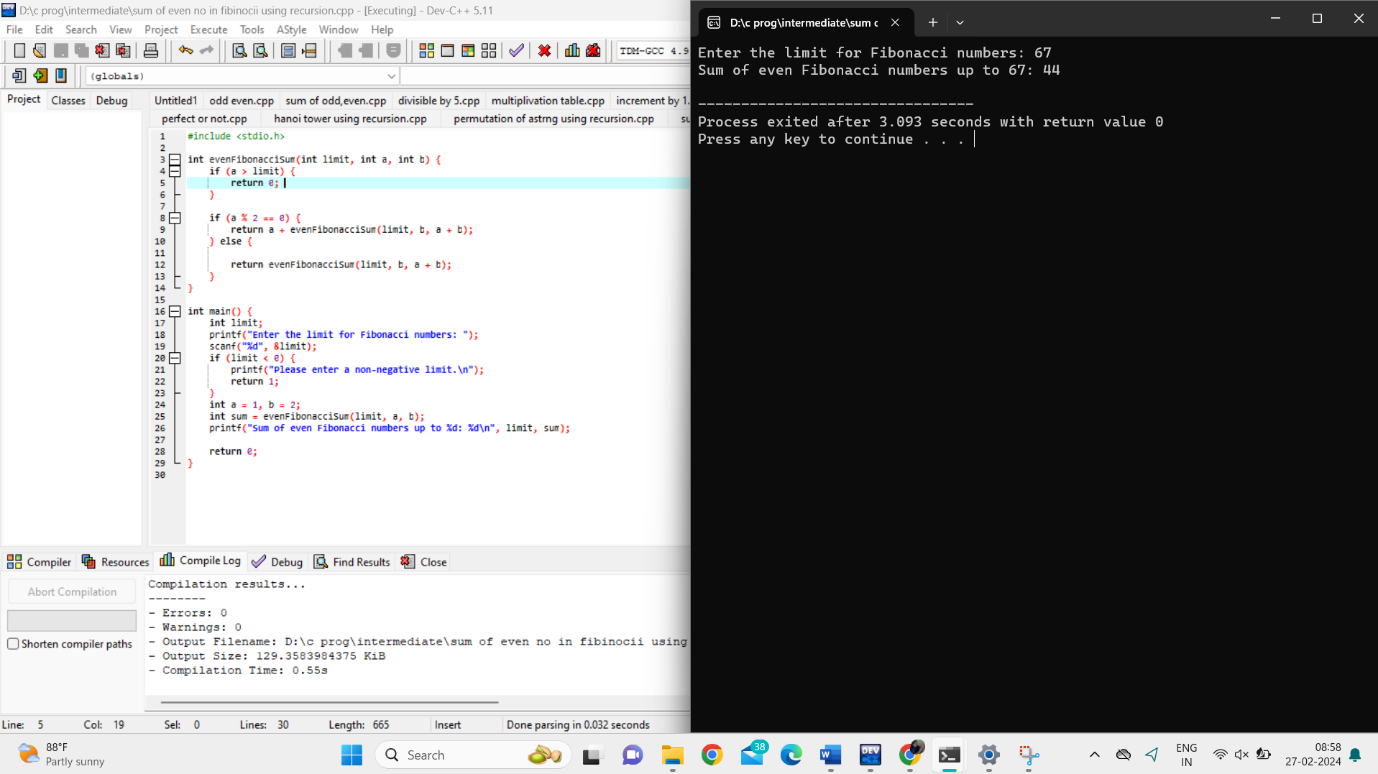
1. Write a program to recursively calculate the sum of all even numbers in a given range.



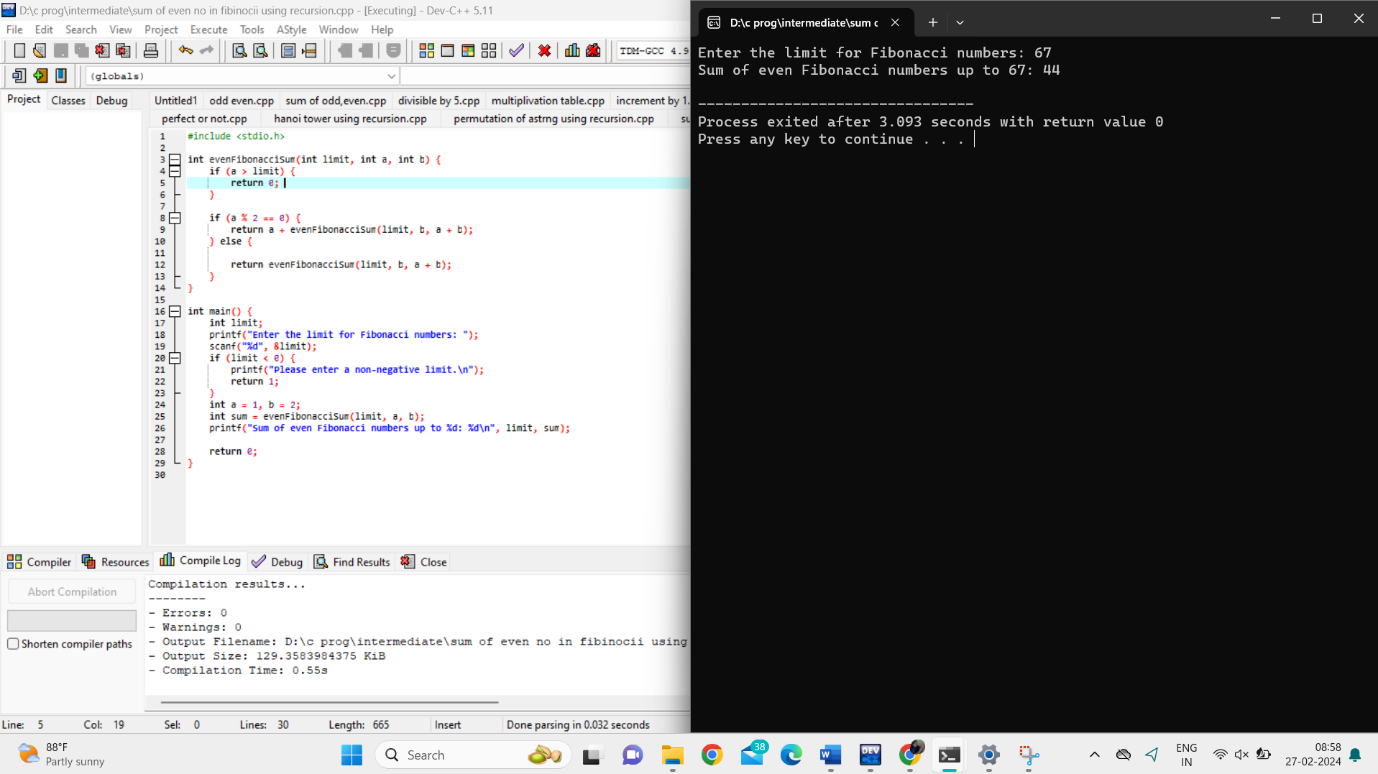
1. Implement a recursive program to find the binary representation of a decimal number.



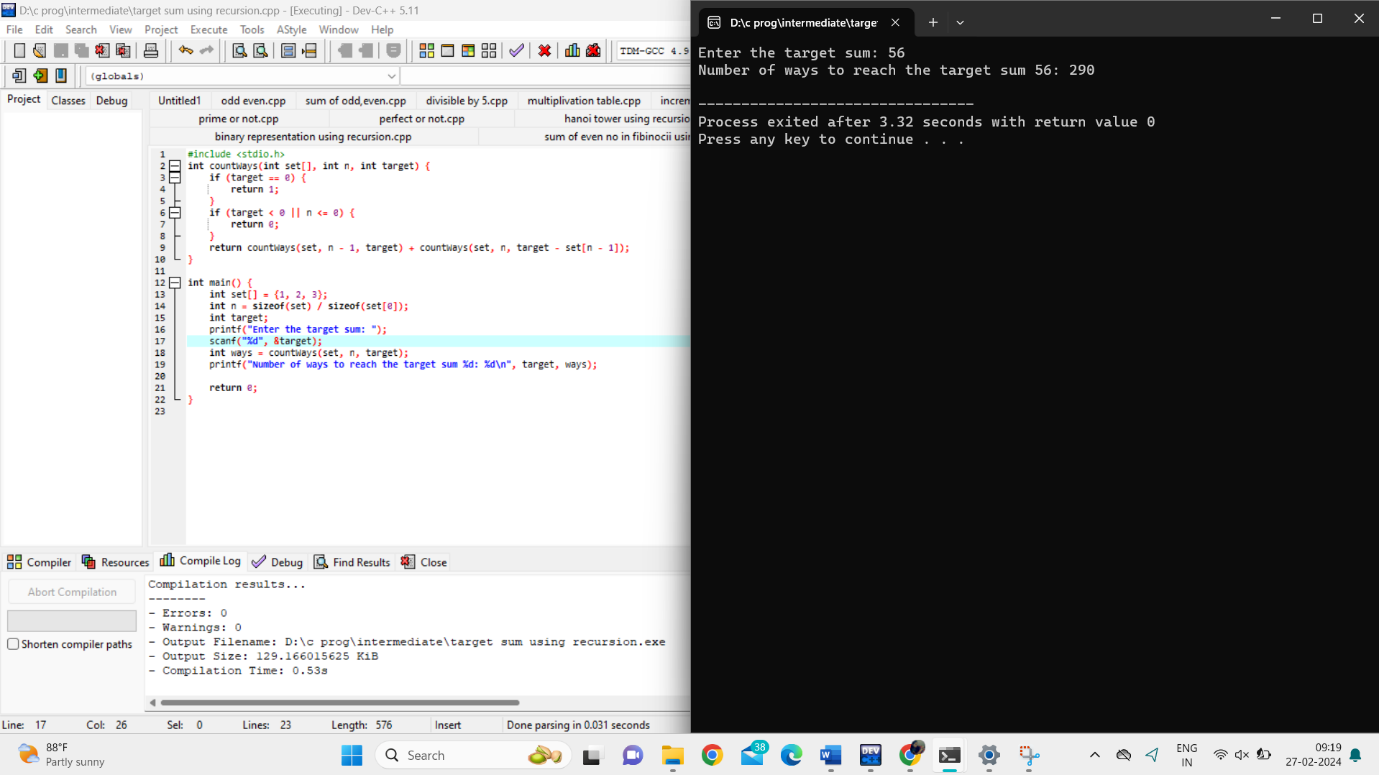
1. Write a program to recursively calculate the sum of even Fibonacci numbers up to a given limit.



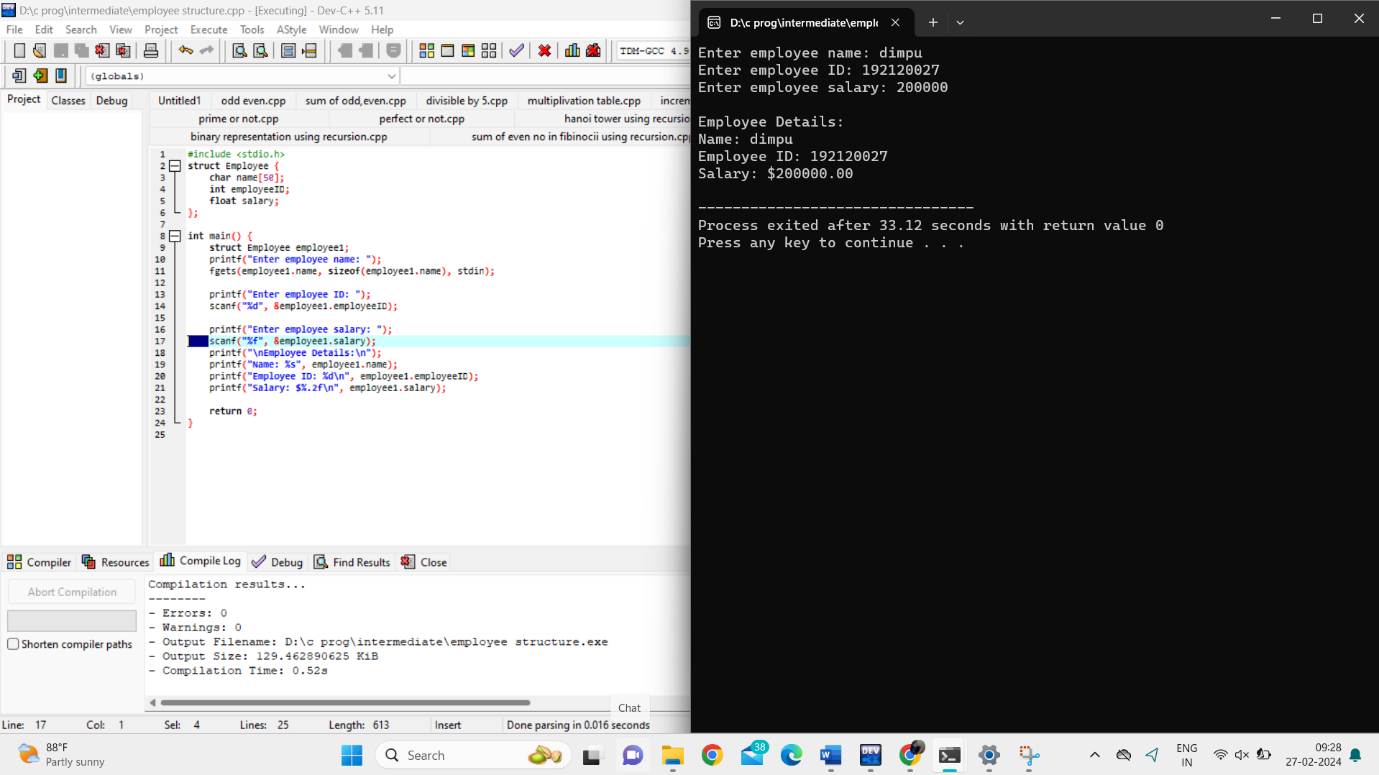
1. Implement a recursive program to solve the Sudoku puzzle.



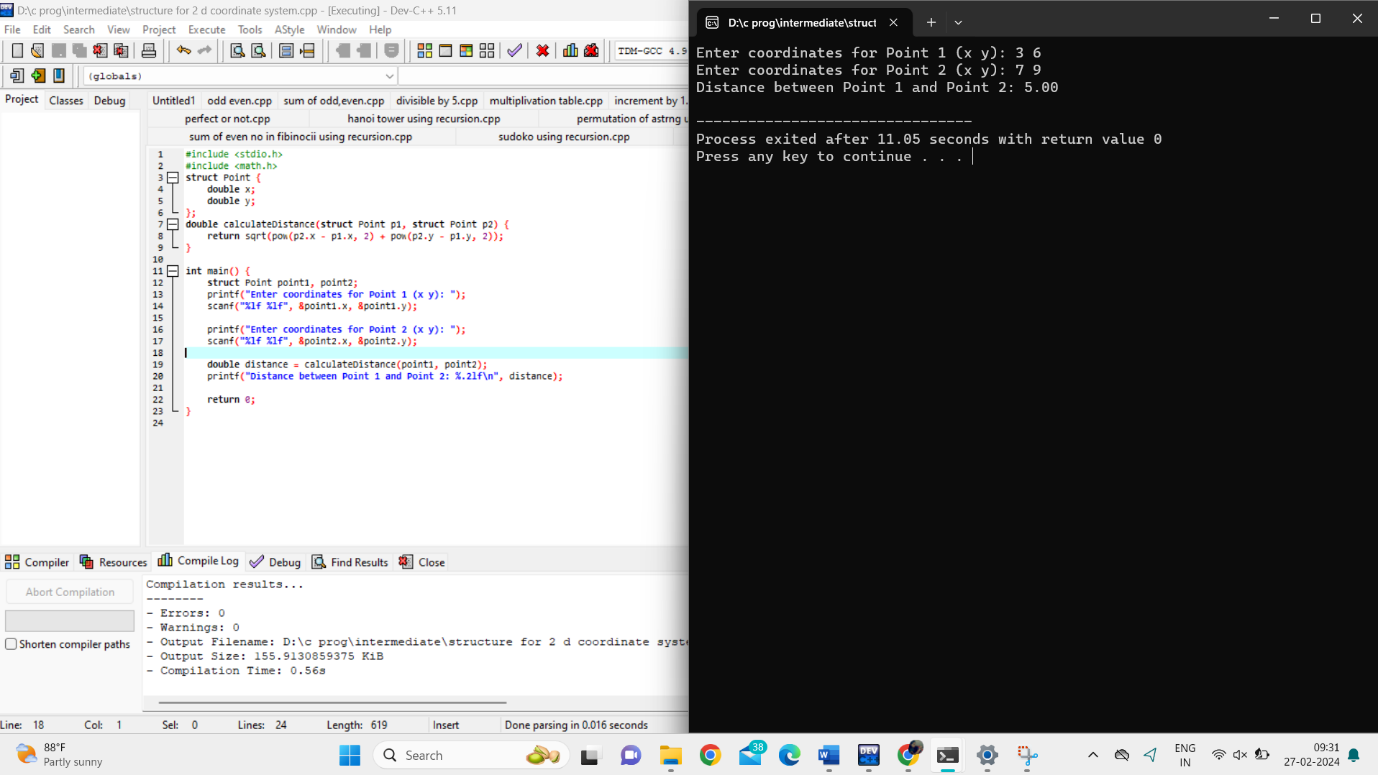
1. Write a program to recursively calculate the number of ways to reach a target sum using a set of given numbers.



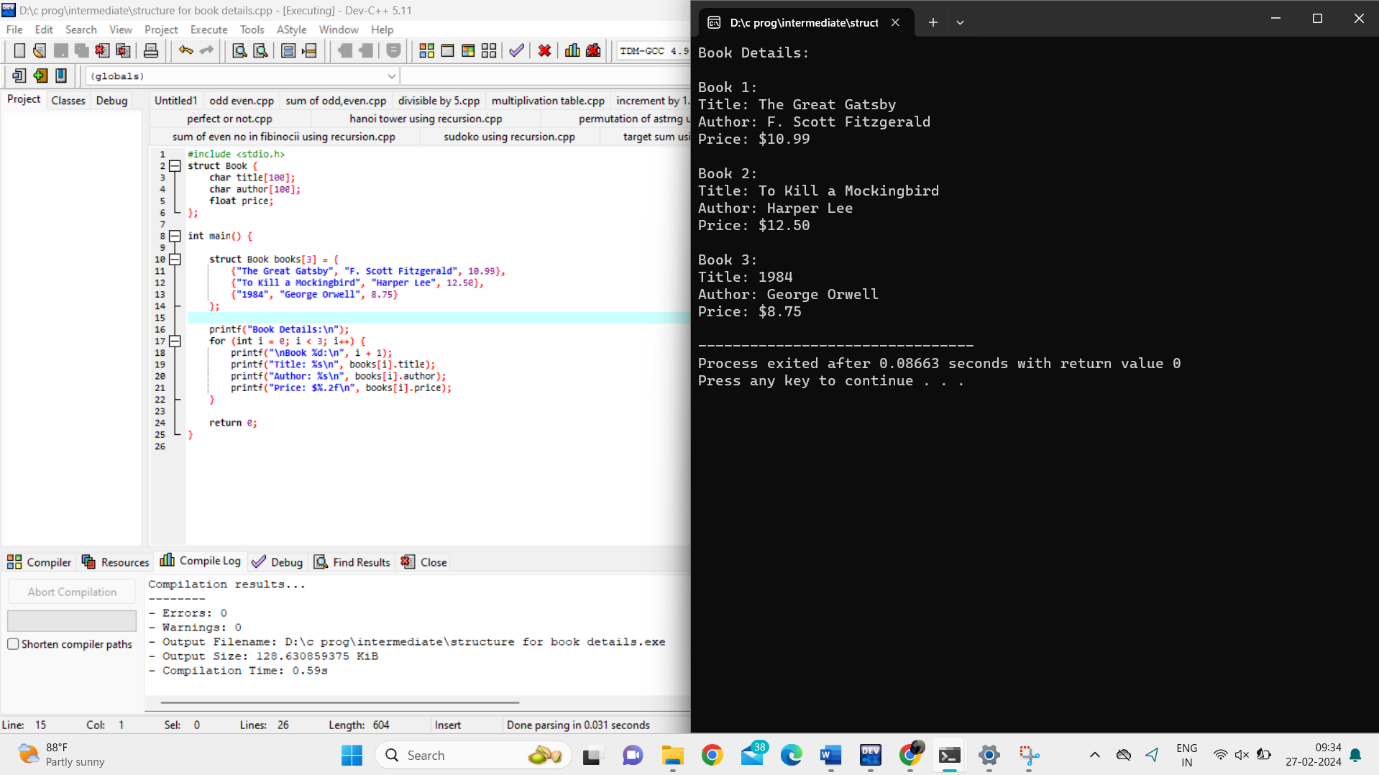
1. Create a structure named "Employee" to store employee details such as name, employee ID, and salary. Write a program to initialize and display the details of an employee using this structure.



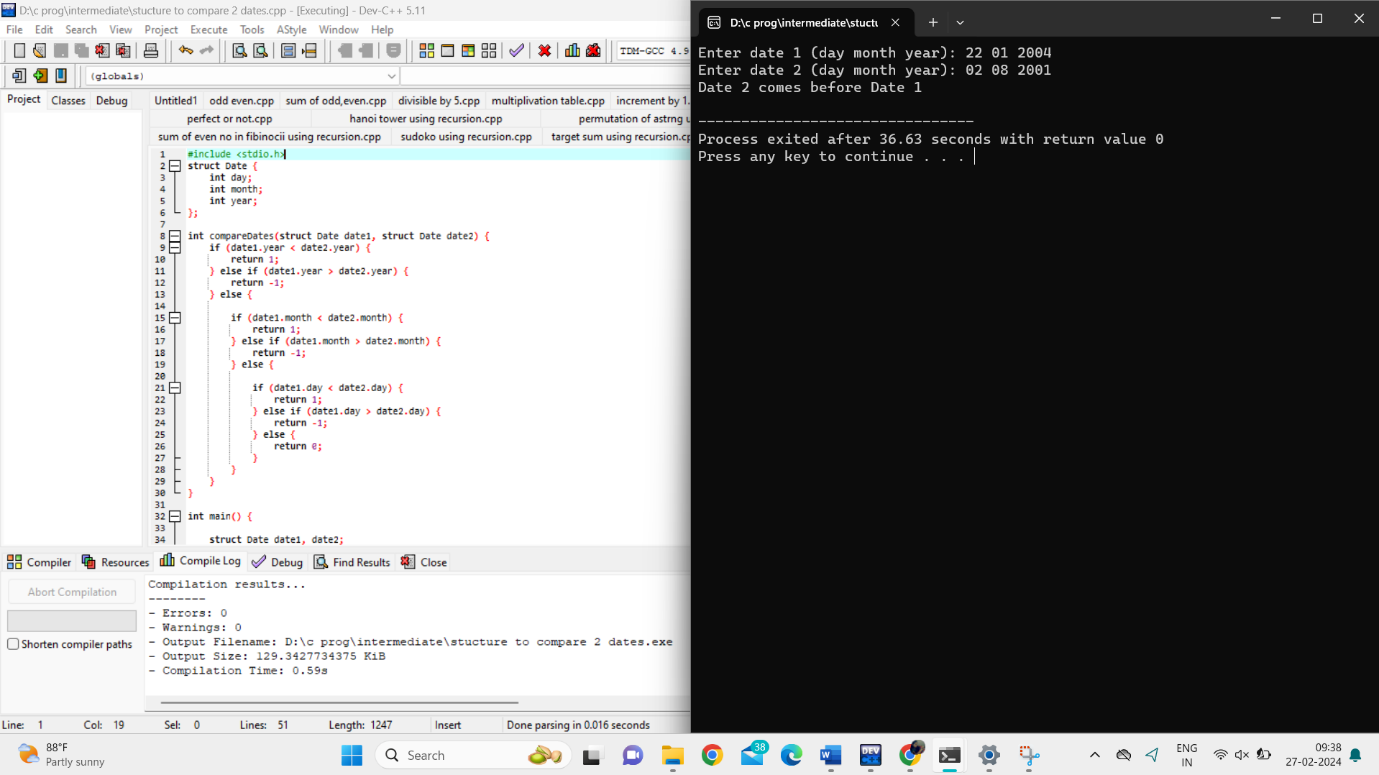
Define a structure named "Point" to represent a point in a 2D coordinate system. Write a program to calculate the distance between two points using this structure.



a structure named "Book" to store book details such as title, author, and price. Write a program to initialize an array of books using this structure and display their details.



1. Define a structure named "Date" to represent a date (day, month, and year). Write a program to compare two dates using this structure and display which date comes first.



Create a structure named "Student" to store student details such as name, roll number, and marks in three subjects. Write a program to calculate the average marks of a student using this structure.

