**Data Structures and Algorithms: Programming Assignment**

**Course:** MBA. Tech CE (Div-A)– Semester III  
**Subject Code:** 702CO1C001  
**Instructor:** Dr. Nitin Choubey  
**Submission Deadline:** 01st November 2025  
**Mode of Submission:** Submit GitHub repo link or PDF with screenshots and code as submission for this.

**Objective**

To strengthen your understanding of core data structures and algorithms by solving real-world problems on competitive programming platforms. This assignment will help you develop problem-solving skills, optimize code, and prepare for technical interviews.

**Platforms**

You may use any of the following platforms:

1. LeetCode
2. CodeChef
3. HackerRank
4. GeeksforGeeks Practice

**Assignment Task: Data Structures Practice**

Solve any **two problems** from the mentioned categories,

1. Array
2. Linked List
3. Stack
4. Queue

**(Note: if you are chosen first problem from the Array, then the second problem should be from the other three categories)**

**Submission Guidelines**

* Create a GitHub repository named DSA\_Assignment\_YourName
* For each problem, include:
  + Problem link
  + Your code (with comments)
  + Screenshot of successful submission
  + Brief explanation of your approach (2–3 lines)
* Alternatively, compile all of the above into a single PDF and upload it to MS Teams.

**Evaluation Criteria**

* Completion of required problems
* Code correctness and efficiency
* Clarity of explanations
* Proper submission format

Name: Tithi Bansal

Roll No: N269

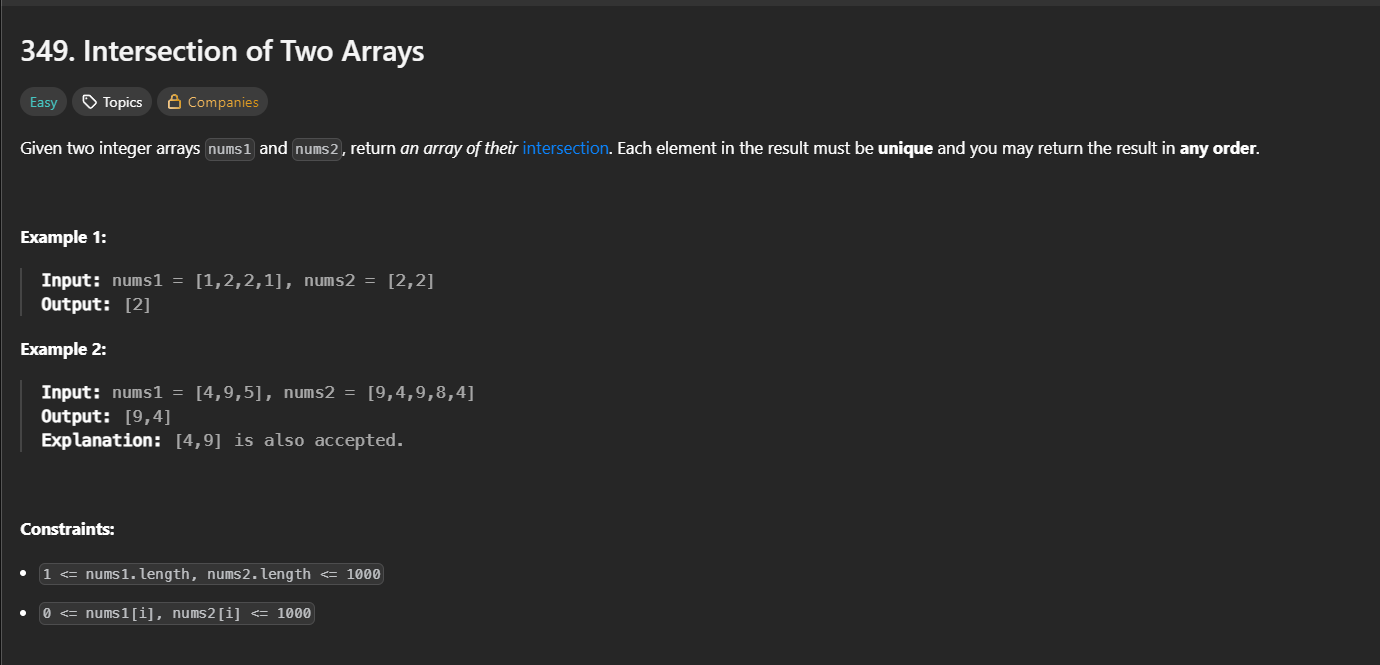
Batch: A2

Class: MBA-TECH(CE)

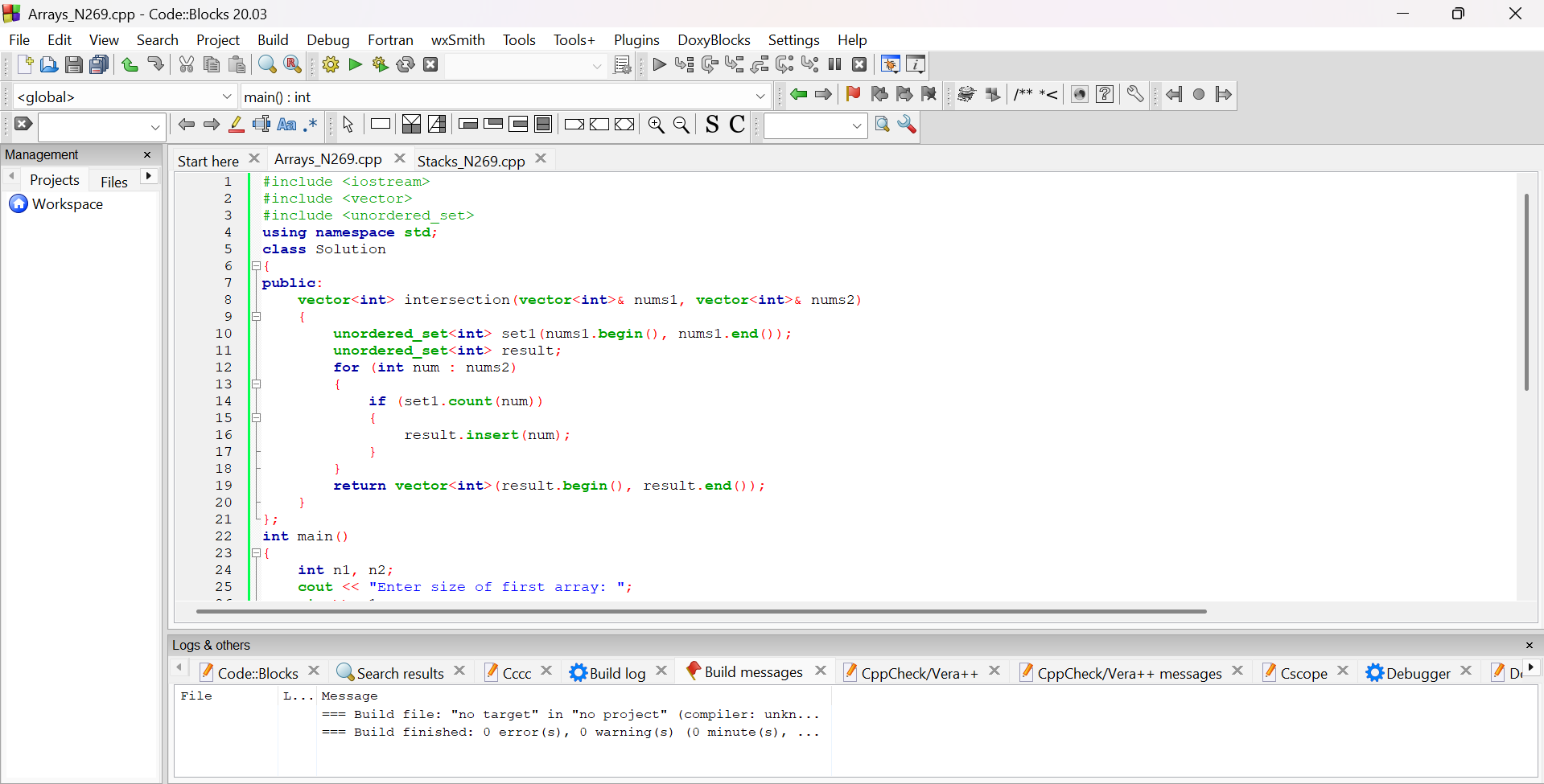
Github link: <https://github.com/tithibansal/DSA_Assignment_Tithi>

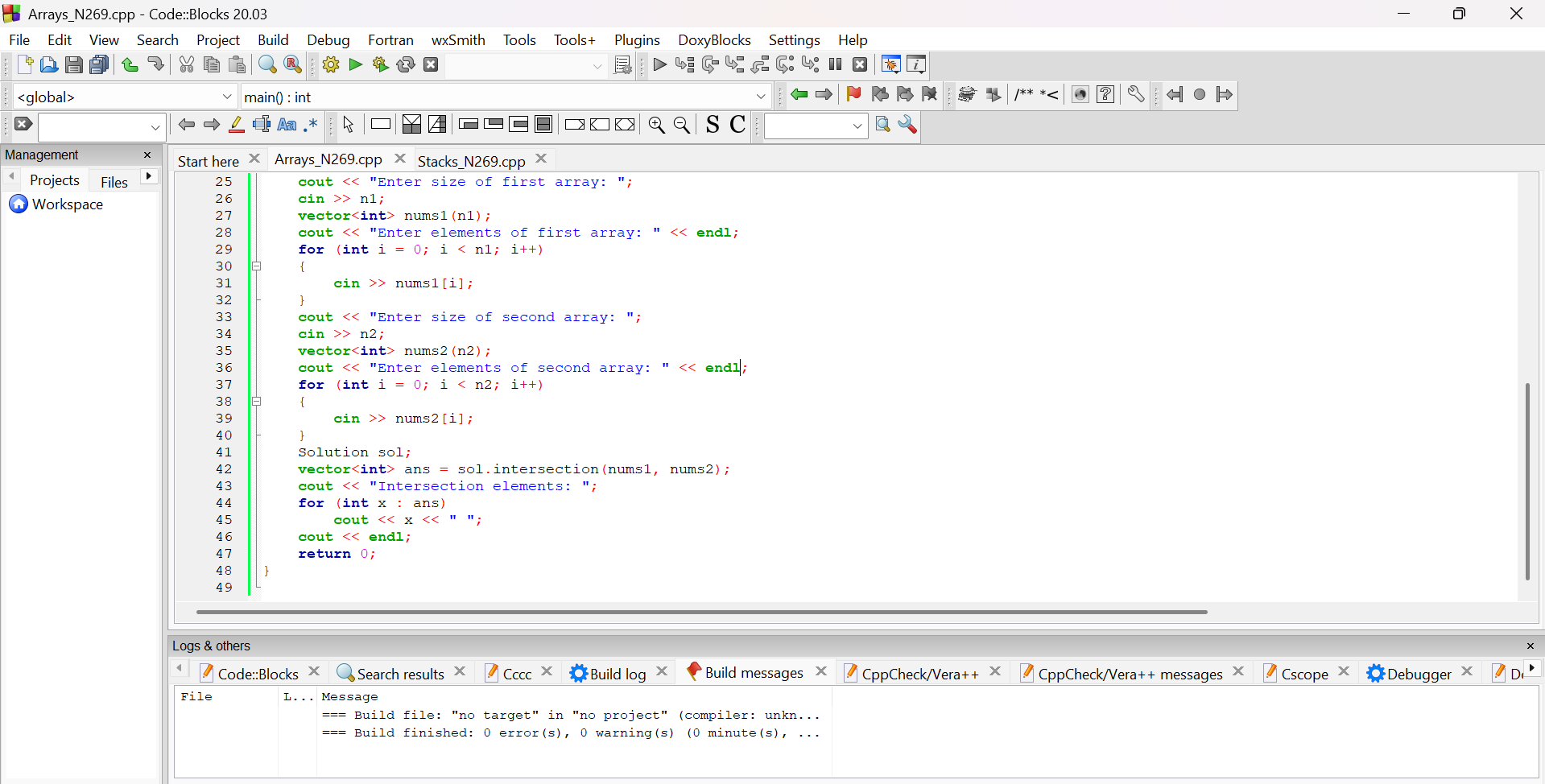
1. Arrays: <https://leetcode.com/problems/intersection-of-two-arrays/>

**PROBLEM STATEMENT**

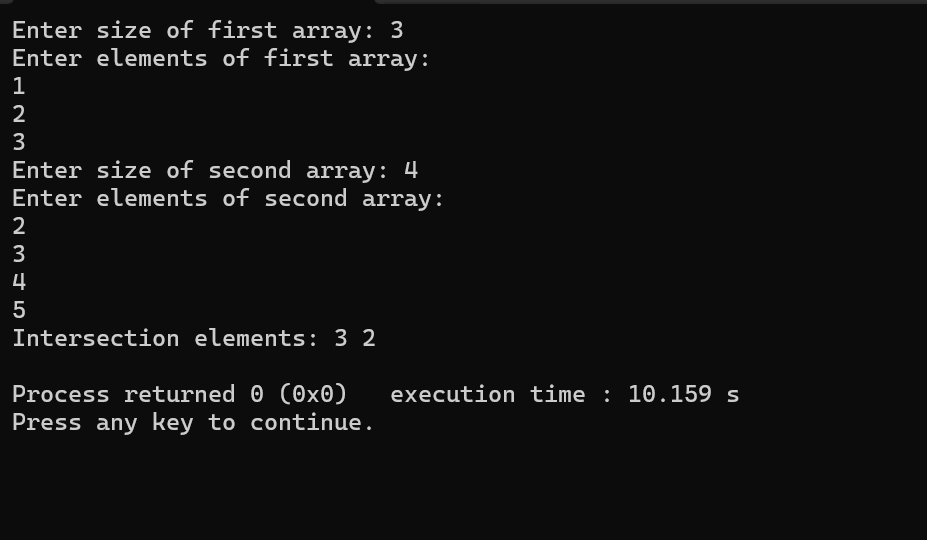


**CODE**





**OUTPUT**



**BRIEF DESCRIPTION ABOUT THE APPROACH**

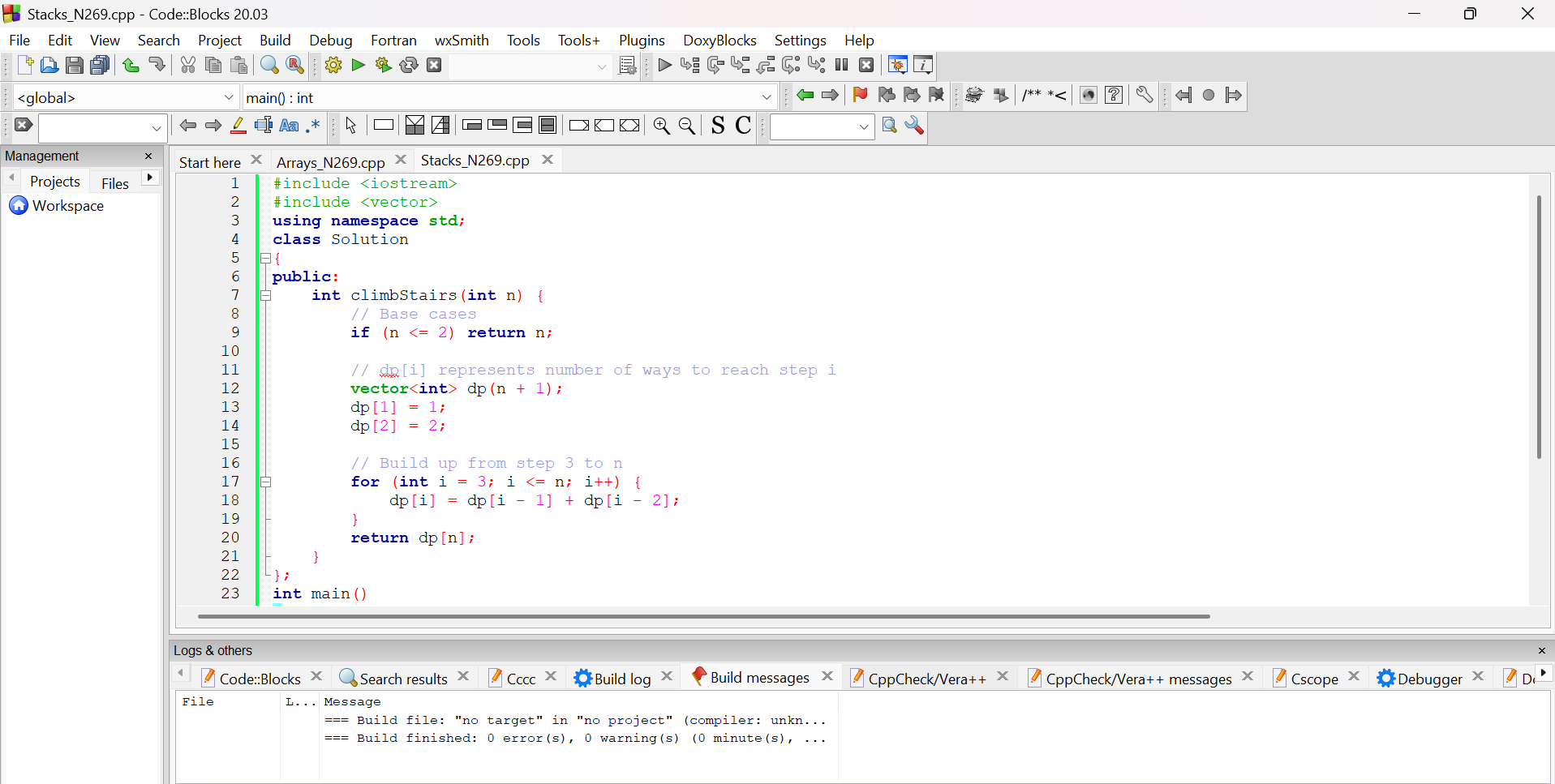
The goal of the problem is to find all the **unique elements** that are common to both input arrays, nums1 and nums2. The approach used in the code is very efficient: it uses an **unordered\_set** (a hash set) to store all the unique elements of the first array, which allows for very fast lookups ($O(1)$ average time). Then, it iterates through the second array, checking if each element is present in the set. If it is, the element is part of the intersection and is added to a **second set** (result) to ensure that the final output only contains **unique** intersection elements. Finally, this second set is converted back into a vector<int> for the return value.

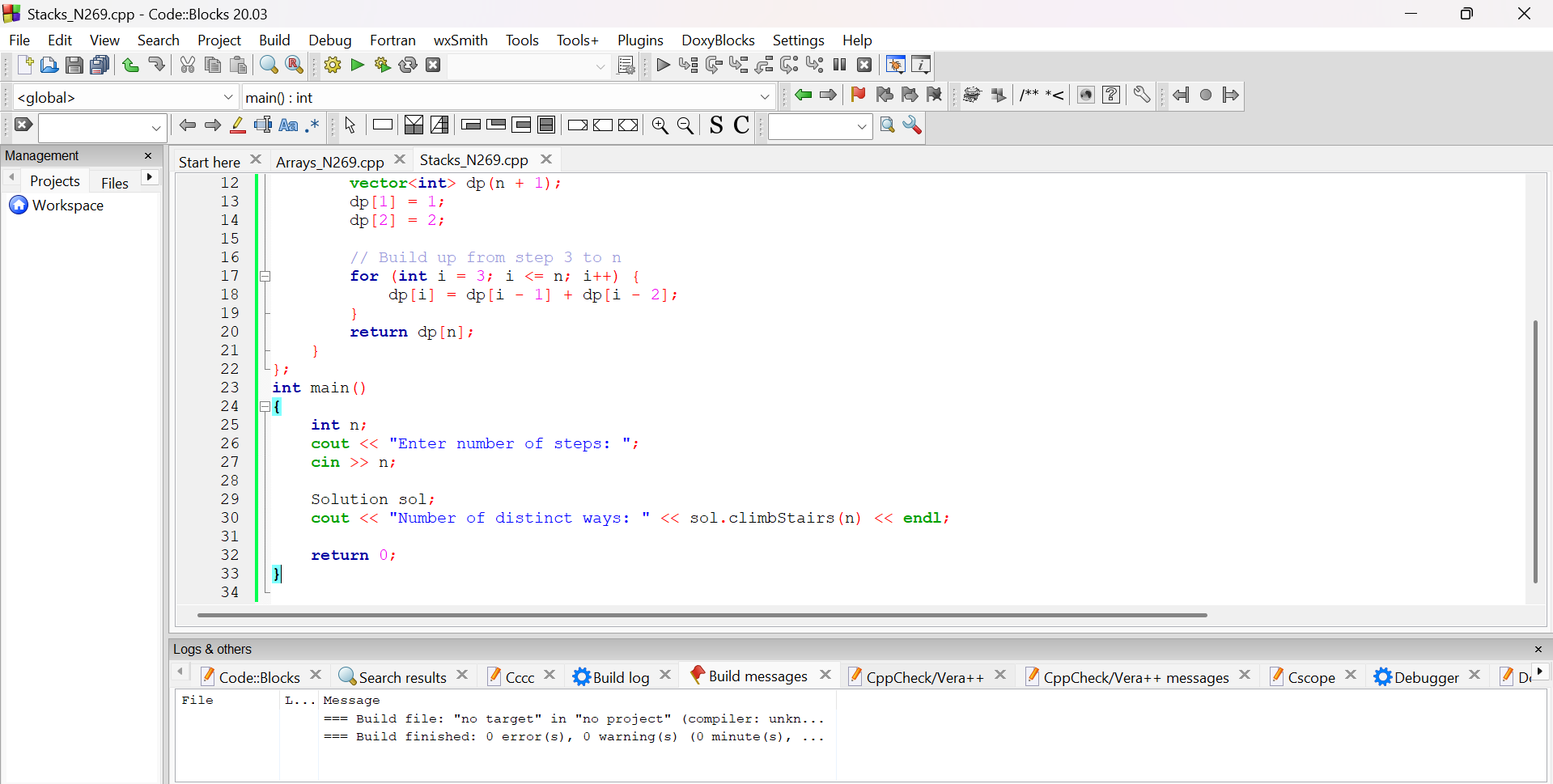
1. Stacks: <https://leetcode.com/problems/climbing-stairs/description/>

**PROBLEM STATEMENT**

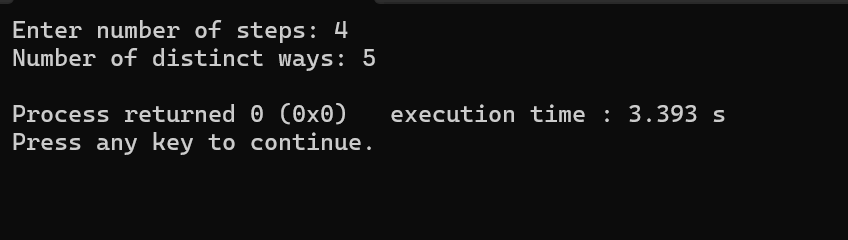


**CODE**





**OUTPUT**



**BRIEF DESCRIPTION ABOUT THE APPROACH**

This code solves the "Climbing Stairs" problem, which asks for the number of distinct ways to climb *n* steps by taking either 1 or 2 steps at a time. The approach uses Dynamic Programming (DP), specifically recognizing that the number of ways to reach step *i* is the sum of ways to reach the previous two steps (*i*−1 and *i*−2), which is the Fibonacci sequence