**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 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

Solution:

Name: Pankti Khakhar

Class: MBATech(CE)-A1

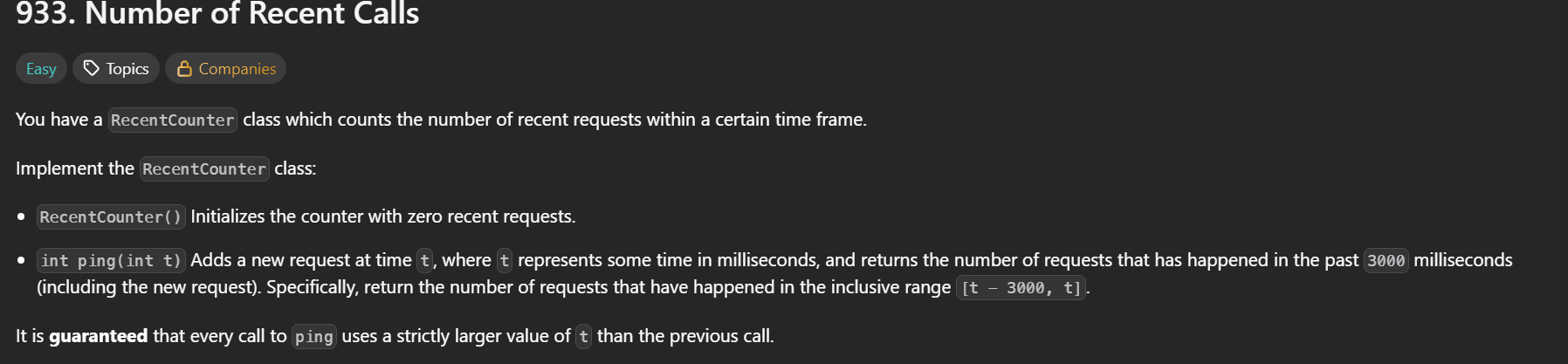
Roll No: N202

Two categories selected: Stacks and Queues

Question-1:

Leetcode Question Link:

<https://leetcode.com/problems/number-of-recent-calls/description/?envType=problem-list-v2&envId=queue>

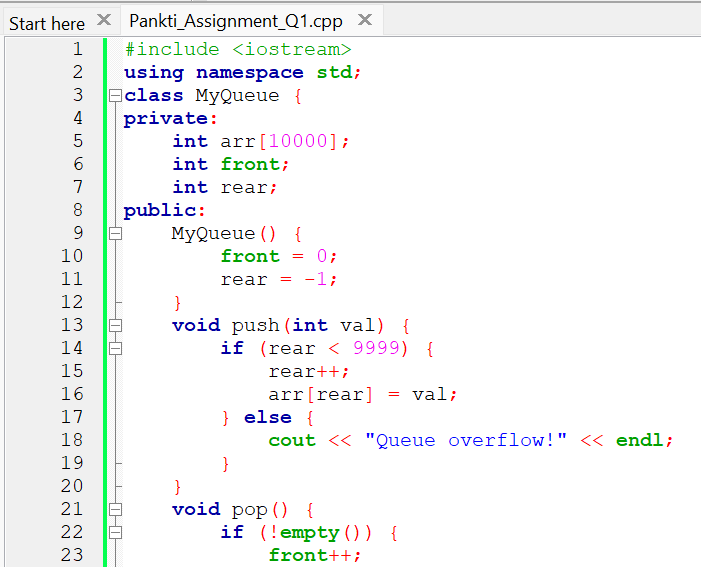


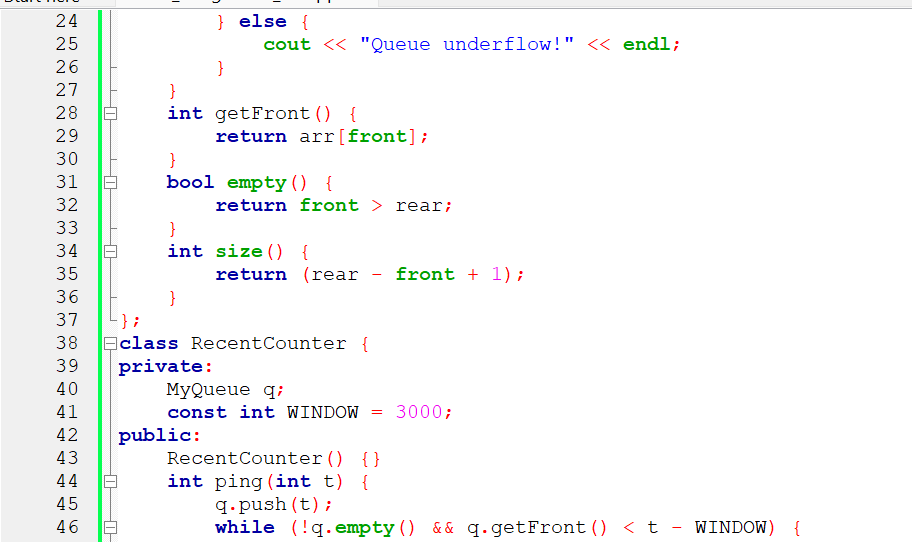
Approach:

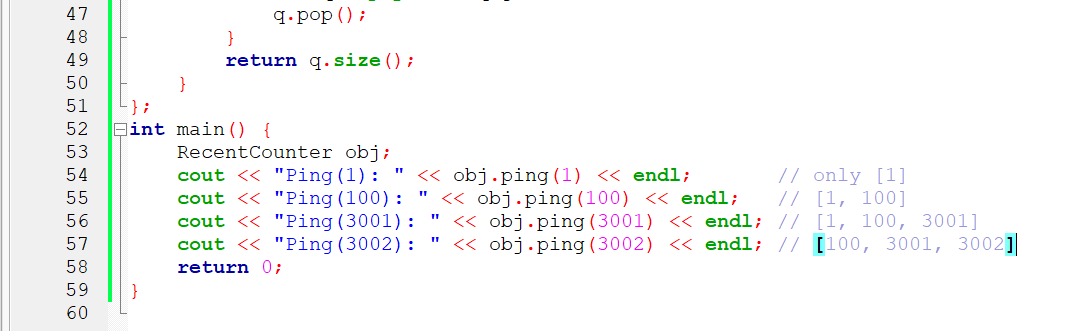
We maintain a queue of timestamps for recent requests.  
 For each new ping t, we add t to the queue and remove all timestamps older than t − 3000.  
 The queue then contains only pings from the last 3000 ms, so its size gives the current count.  
 Time complexity O(1) per call; space O(w), where w is the number of requests in 3000 ms.

Code:

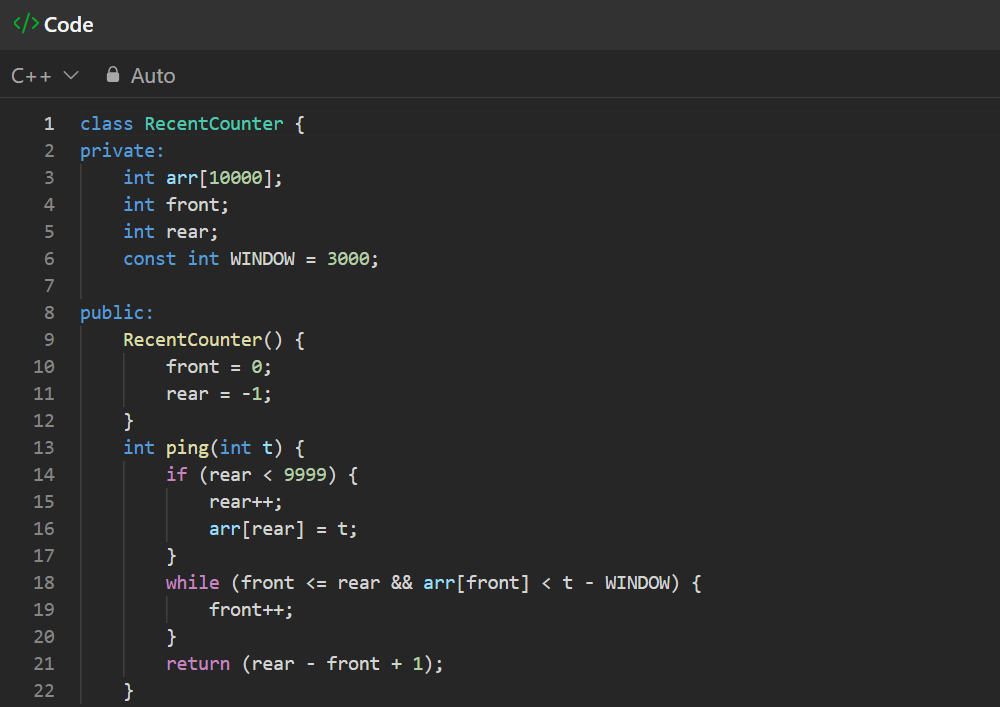
Codeblocks:





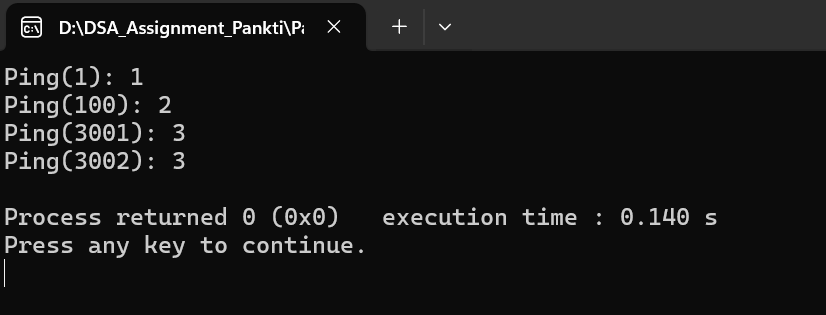


Leetcode:

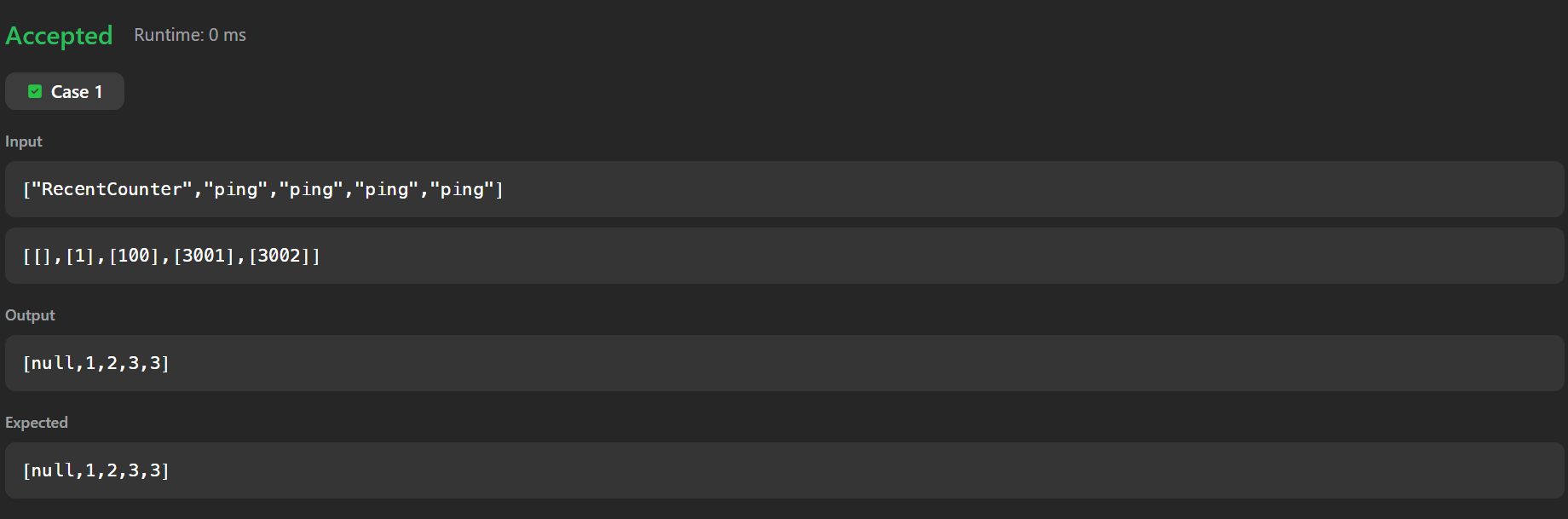


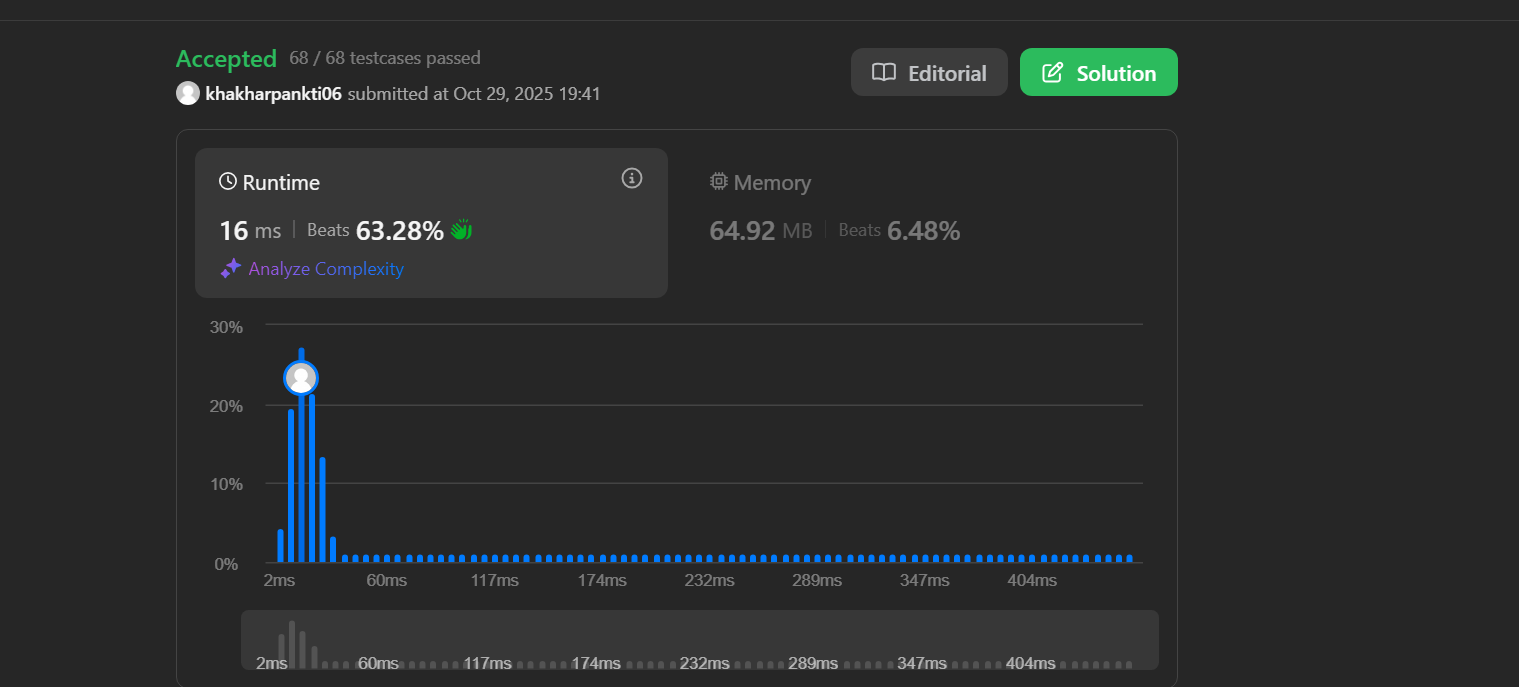
Output:

Codeblocks:



Leetcode:

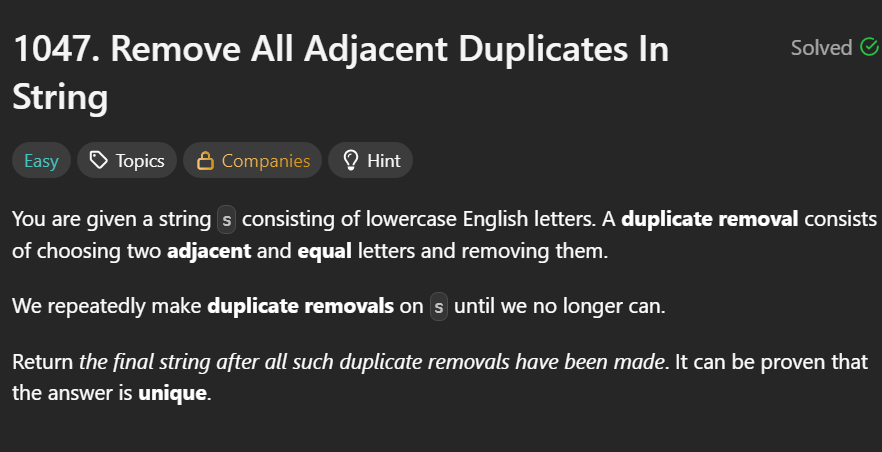




Question-2:

Leetcode Question link:

<https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string/description/?envType=problem-list-v2&envId=stack>



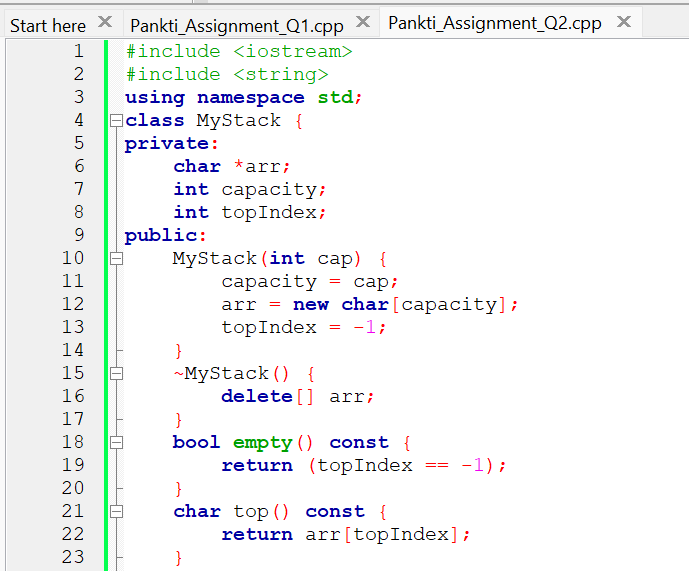
Approach:

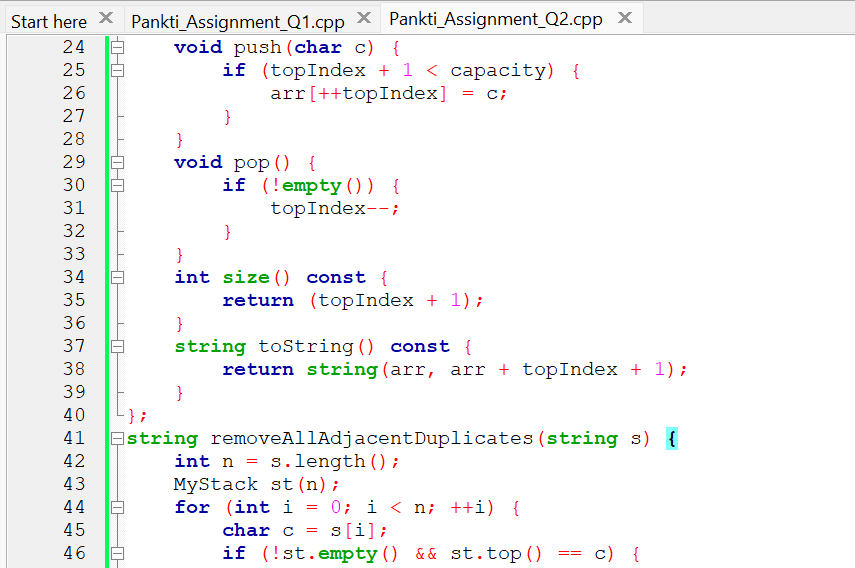
We simulate a stack using an array.  
 Traverse each character of the string:

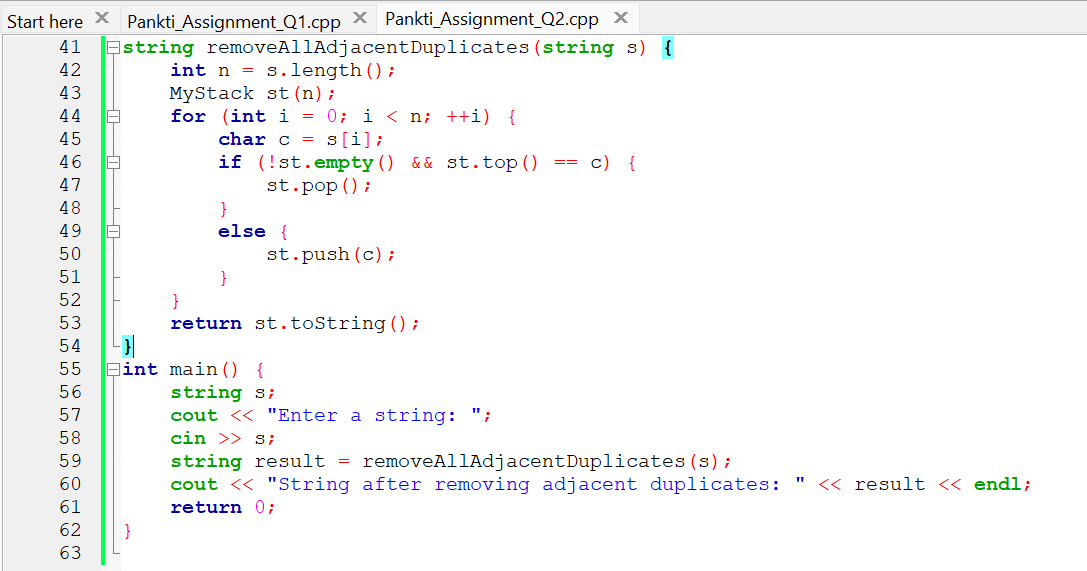
* If the stack isn’t empty and the top equals the current character, pop (remove the pair).
* Otherwise, push the character.  
   After traversal, the stack contents form the final string without adjacent duplicates.  
   Time complexity O(n); space O(n).

Code:

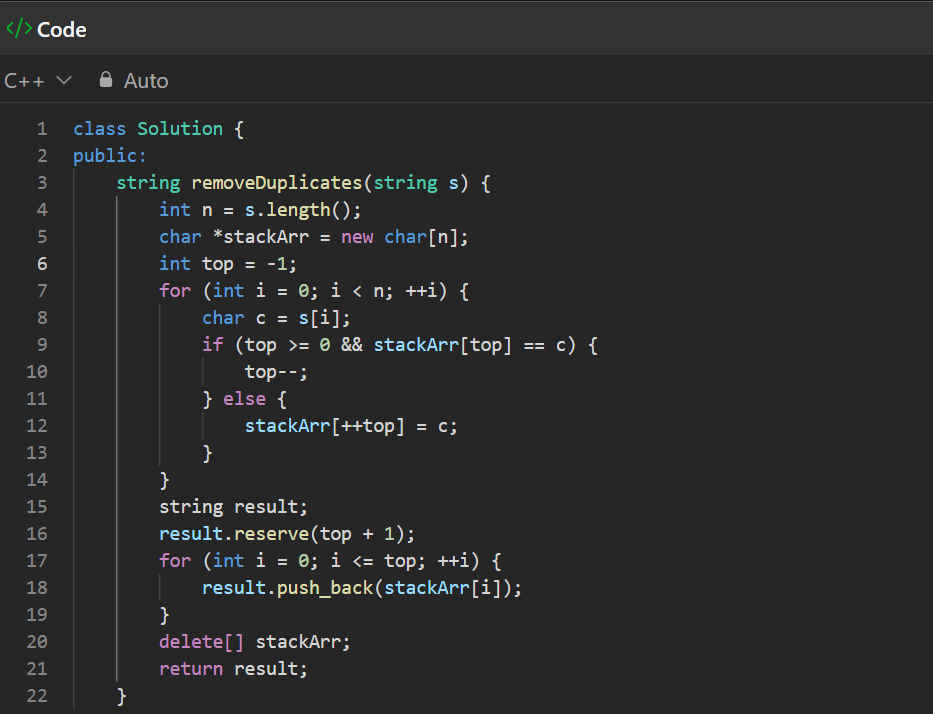
Codeblocks:





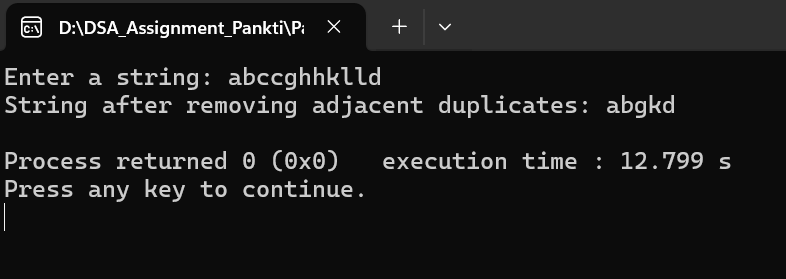


Leetcode:



Output:

Code blocks:



Leetcode:

