**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.

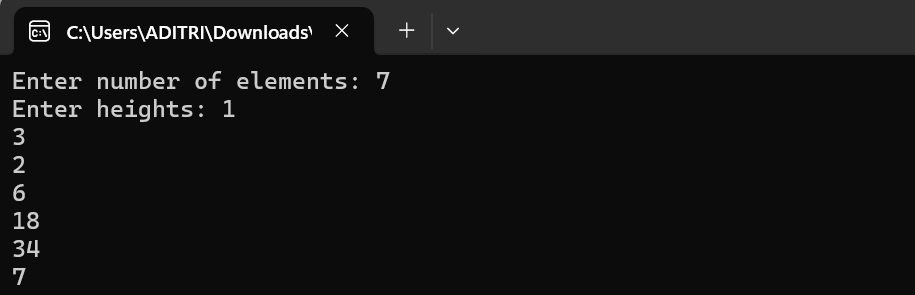
**Evaluation Criteria**

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

**GITHUB LINK:**

**Problem 1**[**:** **https://leetcode.com/problems/container-with-most-water/?envType=problem-list-v2&envId=array**](:%20https:/leetcode.com/problems/container-with-most-water/?envType=problem-list-v2&envId=array)

**11.Container With most water**

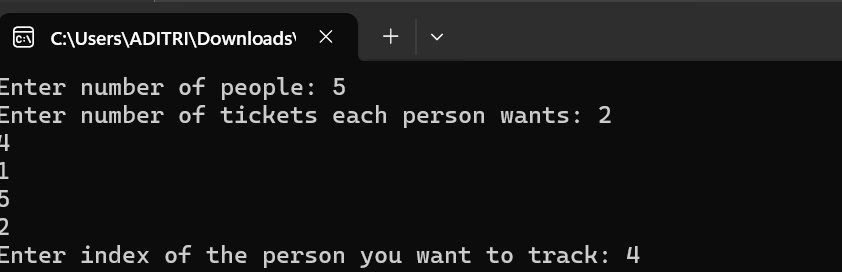
**Input:**

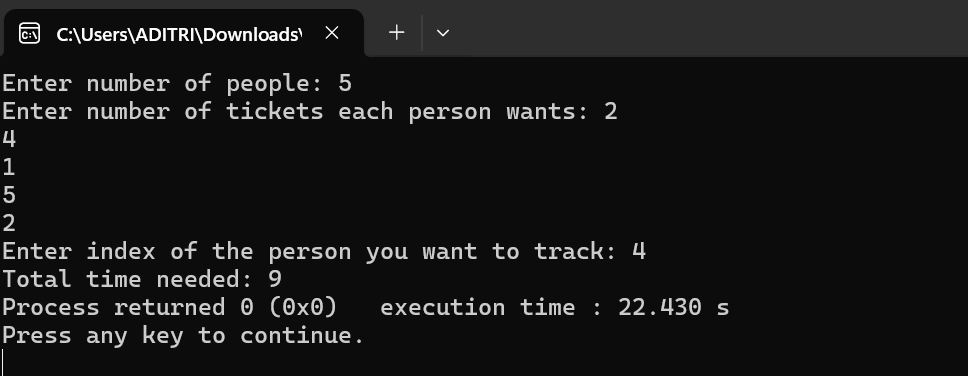
**Output:**

**Brief Explanation:**I used the two-pointer approach where one pointer starts at the beginning and the other at the end of the array. By calculating the water area between them and moving the pointer with the smaller height, we find the maximum area efficiently

**Problem 2:** [**https://leetcode.com/problems/time-needed-to-buy-tickets/description/?envType=problem-list-v2&envId=queue**](https://leetcode.com/problems/time-needed-to-buy-tickets/description/?envType=problem-list-v2&envId=queue)

**2037: Time Needed to Buy Needed**

**Input:**

**Output:**

**Brief Explanation :**I used a queue to simulate people standing in line to buy tickets. Each person buys one ticket per turn, and if they still need more, they move to the back of the queue. The process continues until the target person (k) buys all their tickets.