

# TechnoJam Recruitment Audition 2024-2025

## (DSA Domain)

Dear Candidate,

We appreciate your interest in joining TechnoJam for the 2024-2025 session in the DSA domain. As part of our audition process, you are required to solve five Data Structures and Algorithms (DSA) problems.

### Steps to Follow:

- **Solve the Problems:** Complete each of the five provided DSA problems.
- **Upload Your Solutions:** Create a GitHub repository and upload your solutions.
- **Share Your Repository Link:** Provide the link to your GitHub repository for evaluation.

### Submission Requirements:

Along with your solutions, please include the following information for each problem:

- **Your Approach:** Explain the thought process and steps you took to tackle each problem.
- **Time and Space Complexity:** Analyze the efficiency of your solutions regarding time and space complexity.
- **Challenges Faced:** Describe any difficulties or challenges you encountered during the problem-solving process.

Your submission will be evaluated based on your approach, code quality, and the efficiency of your solutions. Additionally, we will discuss your problem-solving process during the interview.

### Important Notes:

- All solutions must be completed and uploaded before your interview.
- Ensure each solution is well-documented with the required information; incomplete submissions may not be considered.
- Feel free to refer to the provided learning resources if you need to reinforce your understanding of the concepts.

## **Folder Structure:**

You will find separate folders for each programming language: **C, C++, Java, and Python**. Please choose the programming language in which you are most comfortable solving the problems.

## **Questions Overview:**

Each folder contains five questions, with each question consisting of:

- **Driver Code:** A template for you to implement your solution.
- **Test Cases:** Example test cases to help you validate your solution.

## **Instructions:**

- **Select a Programming Language:** Choose one of the provided folders.
- **Implement Your Solution:** Write your code in the designated function in the driver code.
- **Test Your Code:** Use the provided test cases to verify that your solution works as intended.
- **Upload Your Solution:** Once completed, upload your code to your GitHub repository and share the link with us.

## **Good Luck!**

We wish you the best of luck in your audition! Should you have any questions or face any challenges, please feel free to reach out to us.

**TechnoJam**  
*Throttle to Learn*

## Easy level

Q1. Generate the first n rows of Pascal's Triangle.

Input: n = 5

```
1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1
```

Output

Q2. You are climbing a staircase. It takes n steps to reach the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Input: n = 2

Output: 2

Explanation: There are two ways to climb to the top. 1.

i. 1 step + 1 step

ii. 2 steps

TechnoJam  
Throttle to Learn

## Medium

Q3. Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value.

If target is not found in the array, return [-1, -1].

You must write an algorithm with  $O(\log n)$  runtime complexity.

**Example 1:**

**Input:** nums = [5,7,7,8,8,10], target = 8

**Output:** [3,4]

**Example 2:**

**Input:** nums = [5,7,7,8,8,10], target = 6

**Output:** [-1,-1]

**Example 3:**

**Input:** nums = [],target = 0

**Output:** [-1,-1]

**TechnoJam**  
*Throttle to Learn*

## String-compression

Q4. Given an array of characters chars, compress it using the following algorithm:

Begin with an empty string s. For each group of **consecutive repeating characters** in chars:

- If the group's length is 1, append the character to s.
- Otherwise, append the character followed by the group's length.

The compressed string s should not be returned separately, but instead, be stored **in the input character array chars**. Note that group lengths that are 10 or longer will be split into multiple characters in chars.

After you are done **modifying the input array**, return *the new length of the array*.

You must write an algorithm that uses only constant extra space.

### Example 1:

**Input:** chars = ["a","a","b","b","c","c","c"]

**Output:** Return 6, and the first 6 characters of the input array should be:

["a","2","b","2","c","3"]

**Explanation:** The groups are "aa", "bb", and "ccc". This compresses to "a2b2c3".

### Example 2:

**Input:** chars = ["a"]

**Output:** Return 1, and the first character of the input array should be: ["a"]

**Explanation:** The only group is "a", which remains uncompressed since it's a single character.

### Example 3:

**Input:** chars = ["a","b","b","b","b","b","b","b","b","b","b"]

**Output:** Return 4, and the first 4 characters of the input array should be: ["a","b","1","2"].

**Explanation:** The groups are "a" and "bbbbbbbbbbb". This compresses to "ab12".

Hard

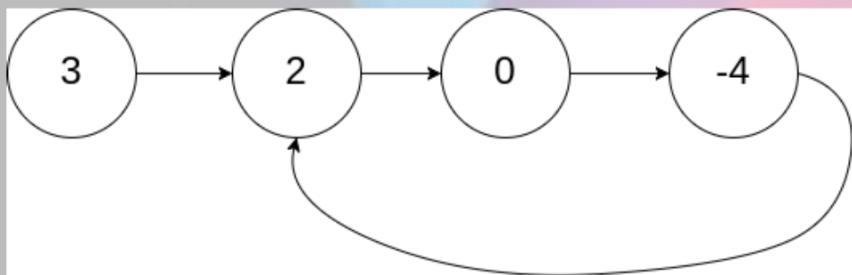


Q5. Given the **head** of a linked list, return *the node where the cycle begins. If there is no cycle, return null.*

There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the **next** pointer.  
Internally, **pos** is used to denote the index of the node that tail's **next** pointer is connected to (**0-indexed**). It is **-1** if there is no cycle. **Note that pos is not passed as a parameter.**

Do not modify the linked list.

Do not modify the linked list.



Example 1:

**Input:** head = [3,2,0,-4],

pos = 1

**Output:** tail connects to node index 1.

**Explanation:** There is a cycle in the linked list, where tail connects to the second node.

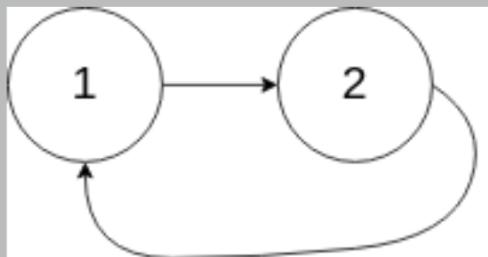
**TechnoJam**  
Throttle to Learn

Example 2:

**Input:** head = [3,2,0,-4], pos = 1.

**Output:** tail connects to node index 1.

**Explanation:** There is a cycle in the linked list, where tail connects to the



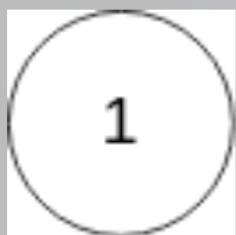
second node.

**Example 3:**

**Input:** head = [1], pos = -1

**Output:** no cycle.

**Explanation:** There is no cycle in the linked list.



**TechnoJam**  
Throttle to Learn