Technical Test

Test Instructions :

- Create A git repository named “lamprostechTest[your name without space]”. For example, Ravi Sharma can create it as “lamprostechTestRaviSharma”.
- You can use any language that you prefer to work on.
- Make sure you comment on the code with the logic you are using.
- If you think you are not able to perform the practical due to time constraints, write your logic in the comments and try to complete it as much as possible.
- Unless it is absolutely necessary, you should not utilise built-in functions like sort.
- Once you have completed the assignment, push the files in the same repo with your solution.
- Save your output screenshots with the name “program_name_solution”.
- Share your GitHub repository and output (results) screenshots by filling out this form: https://docs.google.com/forms/d/e/1FAIpQLScF5gQlsyvU8c5M8gopVH50yrz85MlNWd2jqRLV0XibfTETjQ/viewform

---

Question: 1

Course Schedule (Topological Sort):

- Problem Description: Given a list of courses and their prerequisites, find a valid order in which students can take all courses without violating prerequisites.
- Example: Courses A, B, C, D, with prerequisites: A -> B, C -> B, D -> B
- Expected Code Output: A valid course order (e.g., C -> D -> A -> B)
- Hint: Use topological sort to find an ordering where there are no cycles (a student taking a course before fulfilling its prerequisite).

Question: 2

. **Top K Frequent Elements:**

- Problem Description: Given an array of elements with potential duplicates, find the k most frequently occurring elements in the array.
- Example: Array [5, 2, 5, 3, 5, 3, 1] with k = 2 would return [5, 3] as the top 2 frequent elements.
- Expected Code Output: A list of the k most frequent elements.
- Hint: Use a hashmap to count element frequencies and then apply a sorting or heap-based approach to find the top k elements.