National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Advanced Programming**	**Course Code:**	**CS 2000**
Program:	**BS (CS)**	**Semester:**	**Spring 2025**
Due Date	**14 Feb 2025**	**Total Marks:**	**100**
Section:	**8A , 8B**	**Page(s):**	**4**
Type:	**Assignment 1**		

Instructions: For this assignment, each question should be solved in a separate .js file, clearly labeled with the question number (e.g., question1.js). Ensure that your code produces meaningful console outputs that clearly demonstrate the functionality of your solution. Submissions must be turned in on time; late submissions may not be accepted. Any form of plagiarism will result in an automatic grade of 0. Make sure that your code is well-structured and commented, where necessary, to improve readability and showcase your problem-solving approach. You can take reasonable Assumptions in case of any ambiguity.

Question 1:

Write a function flattenArray that takes a deeply nested array of integers and returns a flattened array. For example, given the input [[1, [2, [3]]], 4, [5, [6, [7]]]], the output should be [1, 2, 3, 4, 5, 6, 7]. Implement the function without using Array.prototype.flat().

Question 2:

Write a function rotateArray that rotates an array of integers in place to the right by k steps. For example, given the array [1,2,3,4,5,6,7] and k = 3, the output should be [5,6,7,1,2,3,4]. Do this in O(1) space.

Question 3:

You are tasked with creating a bank account system using concept of closures and lexical scoping to simulate a real-world banking experience.

Understanding lexical scope and closures:

Lexical scope of an item means the place or scope where this item was created. The item can be created in a global scope and used inside a function, the lexical scope of that item would be global scope.

A closure property means a function that remembers its lexical scope, even when the function is executed outside that scope. This means that an inner function can access variables from its outer function even after the outer function has finished executing.

1. **Create a createBankAccount() function**:
    - This function should take an initial deposit amount as an argument and create a bank account objec**t** that contains the users account information.
        1. Balance (number)
        2. History (array of Strings)

2. **The bank account should support the following methods:**
    - **deposit(amount)**: Adds the given amount to the balance. Each deposit should be logged in the history array with the transaction type and amount.
    - **withdraw(amount)**: Subtracts the given amount (plus a 5-unit fee) from the balance if there are sufficient funds.
        1. If the withdrawal amount exceeds the balance, the function should print an error message, "Error: Insufficient balance."
    - **getBalance()**: Returns the current balance of the account.
    - **getHistory()**: Returns the transaction history (array of objects containing type and amount).
    - **resetAccount()**: Resets the account to its initial deposit and clearing the transaction history.

3. **Create a new method called getAccountSummary() using arrow function:**
    - This method should return Users current balance along with number of transactions made.
    - Use **This** operator inside the arrow function for accessing balance and transactions.
    - If this doesn't work provide explanation in comments and correct this function.

4. **The createBankAccount** function will return all these methods (deposit,withdraw,.etc) and you will call these function from main part of your program.

Question 4:

Your company has developed a software system that processes multiple CVs received from candidates. The software formats these CVs into a JSON structure and forwards them for further processing. Your task is to implement an algorithm that filters and processes the list of candidates based on certain criteria and schedules a test for the selected ones.

1. **Filter Candidates**:
    - Each candidate in the list is represented as an object containing the following fields:
        1. name (string)

2. gpa (number, between 0 and 4)
3. skills (array of strings, e.g., ["JavaScript", "Node.js", "React"])
4. applyDate (string or ISO date format)

- Your task is to filter the candidates who:
  1. Have a GPA greater than or equal to 3.0.
  2. Possess at least one skill from the required skillset list (which will be passed as a parameter to your function).

2. **Schedule Test Date**:
   - For the selected candidates, you need to add a new field testDate to their object.
   - The testDate should be 5 days after the candidate's applyDate. Ensure that you calculate the test date correctly, taking into account weekends and working days.
3. **Output**:
   - Your function should return an array of candidates who are selected, with the testDate added.
   - Merge the resulting array into single object using reduce, where keys are name and values are test date.`

Question 5:

You are given an array of objects, each representing a book. Each book object has the following properties:

- title: The title of the book (string)
- author: The author of the book (string)
- pages: The number of pages in the book (number)

Your task is to write three functions:

1. **Named Function:**
   - takes the array of books as input and returns a new array containing only the titles of the books.
2. **Anonymous Function:**
   - An anonymous function assigned to the variable that takes the array of books and an author name as input and returns a new array containing only the books written by that author.
3. **Arrow Function:**
   - An arrow function called that takes the array of books as input and returns the total number of pages across all books.

Question 6:

You are given an object representing a customer and an array of items in their shopping cart.

1. Using object destructuring, create variables `firstName`, `lastName`, and `city` from the `customer` object.
2. Using rest parameters, write a function `calculateTotalPrice` that takes the `cart` array as input and returns the total price of all items in the cart. The function should be able to handle any number of items in the cart.
3. Using array destructuring, create variables `firstItem` and `secondItem` from the `cart` array. Then, use array destructuring with the rest operator to create a new array called `remainingItems` containing the rest of the items in the cart.