Problem 1: Reverse a Doubly Linked List

Given a pointer to the head node of a doubly linked list, reverse the linked list.

Problem 2: Delete a Node from Doubly Linked List

Given a pointer to the head node of a doubly linked list and a key, delete the node containing that key from the list (if it exists).

Problem 3: Insert a Node at a Given Position in a Doubly Linked List

Given a pointer to the head node of a doubly linked list, insert a new node with a given value at a specified position.

Problem 4: Swap Two Nodes of Doubly Linked List

Given a pointer to the head node of a doubly linked list and two keys, swap these two nodes of the list (if these nodes exists in the list).

Problem 5: Find/ All Unique Triplets with Sum Zero

Given an integer array `nums[]`, write a Java program to find all unique triplets (i.e., combinations of three numbers) such that their sum is zero. Your program should meet the following requirements:

- The program should take an integer array `nums[]` as input.
- It should print all unique triplets that sum to zero.
- If no such triplet exists, the program should output an appropriate message.

Problem 6: Count Frequency of Pairs with Difference k

You are tasked with counting how many pairs of integers in an array have a specific difference `k`. Write a Java program that takes an integer array `arr[]` and an integer `k`, and outputs the count of unique pairs where the absolute difference between the pairs is `k`

Problem 7: Find All Pairs with Product k

Write a Java program that finds all unique pairs of integers in an array whose product equals a target value `k`. Your program should:

- Take an integer array `nums[]` and an integer `k` as input.
- Print all unique pairs whose product equals `k`.

- If no such pairs exist, the program should output an appropriate message.

Problem 8: Find All Pairs with Sum Closest to k

You are given an integer array `arr[]` and an integer `k`. Write a Java program that finds all unique pairs of integers in the array such that their sum is closest to `k` without exceeding it. The program should:

- Print all unique pairs whose sum is closest to `k`.
- If no such pairs exist, the program should output an appropriate message.