Solution@24-11-23

279. Perfect Squares

```
import java.util.Scanner;

public class PerfectSquares {

    public static int numSquares(int n) {
        if (n <= 0) {
            return 0;
        }

        // dp[i] represents the least number of perfect square numbers that sum to i.
        int[] dp = new int[n + 1];

        // Initialize the dp array.
        for (int i = 1; i <= n; i++) {
            dp[i] = Integer.MAX_VALUE;
        }

        // Fill the dp array using dynamic programming.
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j * j <= i; j++) {
                dp[i] = Math.min(dp[i], dp[i - j * j] + 1);
            }
        }

        return dp[n];
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Take user input for the integer.
        System.out.print("Enter an integer: ");
        int num = scanner.nextInt();

        // Call the function to get the result.
        int result = numSquares(num);

        // Output the result.
        System.out.println("The least number of perfect square numbers that sum to " + num + " is: " +
result);

        scanner.close();
    }
}
```

Output

```
java -cp /tmp/XY3yz8pM3s PerfectSquares
Enter an integer: 12
The least number of perfect square numbers that sum to 12 is : 3
```

237. Delete Node in a Linked List

```
class ListNode {
  int val;
  ListNode next;

  public ListNode(int val) {
    this.val = val;
  }
}
```

```
public class DeleteNodeInLinkedList {

 public static void deleteNode(ListNode node) {
  // Copy the value of the next node to the current node
  node.val = node.next.val;

  // Skip the next node by updating the next pointer
  node.next = node.next.next;
 }

 public static void printList(ListNode head) {
  ListNode current = head;
  while (current != null) {
   System.out.print(current.val + " ");
   current = current.next;
  }
  System.out.println();
 }

 public static void main(String[] args) {
  // Example usage:
  // Assume you are given a linked list 1 -> 2 -> 3 -> 4
  ListNode head = new ListNode(1);
  head.next = new ListNode(2);
  head.next.next = new ListNode(3);
  head.next.next.next = new ListNode(4);

  // Node to be deleted is the node with value 2
  ListNode nodeToDelete = head.next;

  System.out.println("Original Linked List:");
```

```
    printList(head);

    // Delete the given node
    deleteNode(nodeToDelete);

    System.out.println("Linked List after deleting the node:");
    printList(head);
  }
}
```

Output

```
PS C:\Users\Ajeet\Desktop\java> javac DeleteNodeInLinkedList.java
PS C:\Users\Ajeet\Desktop\java> java DeleteNodeInLinkedList
Original Linked List:
1 2 3 4
Linked List after deleting the node:
1 3 4
```