**Practical: 03**

**AIM:**

Frequency analysis is a fundamental technique used in cryptanalysis, particularly when dealing with substitution ciphers. It exploits the statistical properties of a language, specifically the frequency at which letters or symbols occur, to decipher encrypted messages. In natural languages, certain letters occur more frequently than others. For instance, in English, 'e' is the most commonly used letter. Frequency analysis involves counting the occurrence of letters in a ciphertext and using that information to deduce the most likely substitutions. Perform frequency analysis attacks on ceaser cipher.

**CODE:**

import java.util.\*;

public class FrequencyAnalysisCaesar {

    private static final String ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    public static String decrypt(String cipherText, int shift) {

        StringBuilder plainText = new StringBuilder();

        for (char ch : cipherText.toCharArray()) {

            if (Character.isLetter(ch)) {

                char base = Character.isUpperCase(ch) ? 'A' : 'a';

                plainText.append((char) ((ch - base - shift + 26) % 26 + base));

            } else {

                plainText.append(ch);

            }

        }

        return plainText.toString();

    }

    public static Map<Character, Integer> frequencyAnalysis(String text) {

        Map<Character, Integer> freqMap = new HashMap<>();

        for (char ch : text.toUpperCase().toCharArray()) {

            if (Character.isLetter(ch)) {

                freqMap.put(ch, freqMap.getOrDefault(ch, 0) + 1);

            }

        }

        return freqMap;

    }

    public static void main(String[] args) {

        String cipherText = "Wklv lv d whvw phvvdjh"; // Encrypted using Caesar cipher with shift 3

        Map<Character, Integer> frequencyMap = frequencyAnalysis(cipherText);

        System.out.println("Frequency Analysis of Ciphertext:");

        frequencyMap.entrySet().stream()

                .sorted((a, b) -> b.getValue().compareTo(a.getValue()))

                .forEach(entry -> System.out.println(entry.getKey() + ": " + entry.getValue()));

        System.out.println("\nAttempting to decrypt:");

        String bestDecryption = "";

        for (int shift = 1; shift <= 25; shift++) {

            String decryptedText = decrypt(cipherText, shift);

            System.out.println("Shift " + shift + ": " + decryptedText);

            if (decryptedText.equalsIgnoreCase("This is a test message")) {

                bestDecryption = decryptedText;

            }

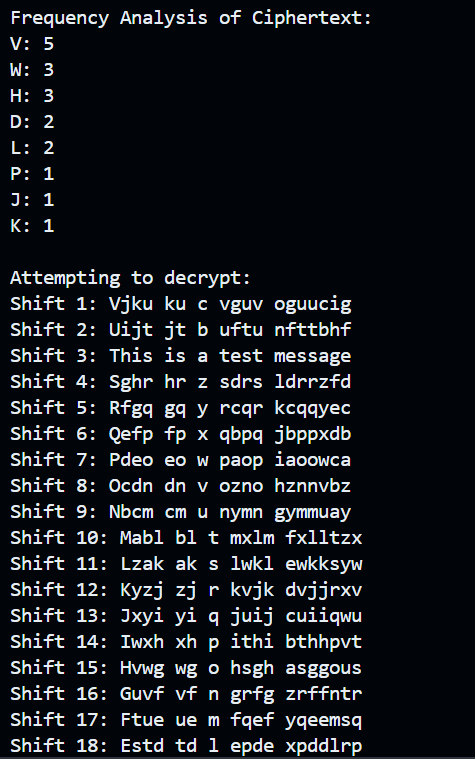
        }

        System.out.println("\nMost likely plaintext: " + bestDecryption);

    }

}

A black screen with white text

AI-generated content may be incorrect.OUTPUT: