**PRACTICAL-2**

**AIM:** Java program that converts a string entered by the user to Morse code or vice versa.

**CODE:**

import java.util.Scanner;

public class Morse {

    // Arrays for letters and corresponding Morse code

    private static final String[] LETTERS = {

        "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M",

        "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "1",

        "2", "3", "4", "5", "6", "7", "8", "9", "0"

    };

    private static final String[] MORSE\_CODE = {

        ".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", ".---", "-.-", ".-..",

        "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-", "..-", "...-", ".--", "-..-",

        "-.--", "--..", ".----", "..---", "...--", "....-", ".....", "-....", "--...", "---..", "----.", "-----"

    };

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Choose conversion type:");

        System.out.println("1. String to Morse code");

        System.out.println("2. Morse code to String");

        int choice = scanner.nextInt();

        scanner.nextLine();  // Consume newline

        if (choice == 1) {

            System.out.println("Enter a string to convert to Morse code:");

            String input = scanner.nextLine().toUpperCase();

            System.out.println("Morse code: " + convertStringToMorse(input));

        } else if (choice == 2) {

            System.out.println("Enter Morse code to convert to string (use space to separate characters):");

            String input = scanner.nextLine();

            System.out.println("String: " + convertMorseToString(input));

        } else {

            System.out.println("Invalid choice.");

        }

        scanner.close();

    }

    private static String convertStringToMorse(String input) {

        StringBuilder morseCode = new StringBuilder();

        for (char c : input.toCharArray()) {

            if (c == ' ') {

                morseCode.append("   ");  // Three spaces to separate words

            } else {

                String morse = findMorseCodeForLetter(c);

                if (morse != null) {

                    morseCode.append(morse).append(" ");

                } else {

                    morseCode.append("? ");  // Unknown character

                }

            }

        }

        return morseCode.toString().trim();

    }

    private static String convertMorseToString(String input) {

        StringBuilder result = new StringBuilder();

        String[] words = input.trim().split("   ");  // Split by three spaces for words

        for (String word : words) {

            String[] letters = word.split(" ");

            for (String letter : letters) {

                String character = findLetterForMorseCode(letter);

                if (character != null) {

                    result.append(character);

                } else {

                    result.append("?");  // Unknown Morse code

                }

            }

            result.append(" ");

        }

        return result.toString().trim();

    }

    private static String findMorseCodeForLetter(char letter) {

        for (int i = 0; i < LETTERS.length; i++) {

            if (LETTERS[i].charAt(0) == letter) {

                return MORSE\_CODE[i];

            }

        }

        return null;

    }

    private static String findLetterForMorseCode(String morse) {

        for (int i = 0; i < MORSE\_CODE.length; i++) {

            if (MORSE\_CODE[i].equals(morse)) {

                return LETTERS[i];

            }

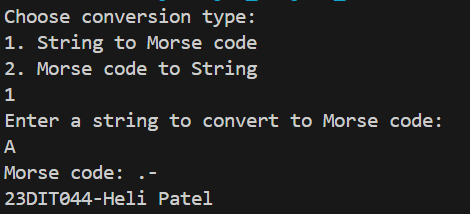
        }

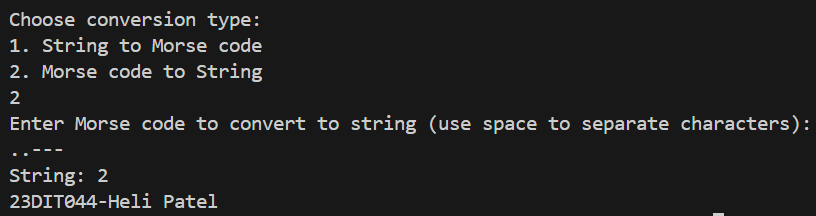
        return null;

    }

}

**OUTPUT:**

****

****