# Braille Autocorrect & Suggestion System

## Problem Statement

Build an autocorrect and suggestion system that:

- Takes Braille dot input via QWERTY keys entered as sets for each Braille character.
- Converts each key set to its corresponding Braille dot pattern.
- Translates the dot patterns into English characters.
- Uses a dictionary to suggest the closest valid word to the typed input.

## Approach

### 1. User Input (QWERTY Braille Keys)

- Users enter Braille characters sequentially.
- For each character, they press keys from the set {`D, W, Q, K, O, P`} *simultaneously*, separated by spaces.
- Key ⇒ dot mapping:

$$D \to 1,\ W \to 2,\ Q \to 3,\ K \to 4,\ O \to 5,\ P \to 6$$

- Hitting *Enter* on an empty line ends input.

### 2. Convert Keys to Dot Pattern (`convertBrailleToText()`)

a) Map each key to its dot number (`Map<Character,Integer>`).
b) Collect dots in a `Set` (order-independent).
c) Sort and join with hyphens (e.g. "1-2-4").
d) Look up the English character in `brailleToChar`.
e) Concatenate characters to build the typed word.

### 3. Suggest Closest Word (`suggestWord()`)

- Compare the typed word with every word in a predefined `List<String>` dictionary.

- Compute Levenshtein distance (`levenshteinDistance()`) for similarity.
- Return the dictionary word with the minimum edit distance.

### 4. Compute Edit Distance (`levenshteinDistance()`)

Dynamic-programming matrix calculates the fewest insertions, deletions, or substitutions needed—tolerating minor mistakes.

### 5. Output

- Display the interpreted word.
- Display the closest dictionary suggestion.

## Technology Stack

- **Language:** Java
- **Core Data Structures:**
  - `Map<Character,Integer>` – QWERTY → dot mapping
  - `Map<String,Character>` – dot pattern → letter mapping
  - `List<String>` – dictionary of words
  - `Set<Character>` – per-character dot collection

## Sample Input / Output

Console Session

**User Input**

```
Enter keys for one Braille character (or press Enter to finish):  D W
Enter keys for one Braille character (or press Enter to finish):  Q K
Enter keys for one Braille character (or press Enter to finish):  D W Q
Enter keys for one Braille character (or press Enter to finish):
```

**Output**

```
Typed word:  bat
Suggested word:  bat
```