**Task 2: Trie for Prefix Checking**

**Implement a trie data structure in java that supports insertion of strings and provides a method to check if a given string is a prefix of any word in the trie.**

**import** java.util.HashMap;

**import** java.util.Map;

**class** TrieNode {

**public** Map<Character, TrieNode> children;

**public** **boolean** isEndOfWord;

**public** TrieNode() {

children = **new** HashMap<>();

isEndOfWord = **false**;

}

}

**public** **class** Trie {

**private** TrieNode root;

**public** Trie() {

root = **new** TrieNode();

}

// Method to insert a word into the Trie

**public** **void** insert(String word) {

TrieNode currentNode = root;

**for** (**char** c : word.toCharArray()) {

currentNode.children.putIfAbsent(c, **new** TrieNode());

currentNode = currentNode.children.get(c);

}

currentNode.isEndOfWord = **true**;

}

// Method to check if a given prefix exists in the Trie

**public** **boolean** startsWith(String prefix) {

TrieNode currentNode = root;

**for** (**char** c : prefix.toCharArray()) {

currentNode = currentNode.children.get(c);

**if** (currentNode == **null**) {

**return** **false**;

}

}

**return** **true**;

}

**public** **static** **void** main(String[] args) {

Trie trie = **new** Trie();

// Inserting words into the Trie

trie.insert("apple");

trie.insert("app");

trie.insert("application");

trie.insert("banana");

// Checking if certain prefixes exist

System.***out***.println(trie.startsWith("app")); // True

System.***out***.println(trie.startsWith("ban")); // True

System.***out***.println(trie.startsWith("cat")); // False

System.***out***.println(trie.startsWith("appl")); // True

System.***out***.println(trie.startsWith("appla")); // False

}

}