```
class Trie {

    class TrieNode {
        TrieNode[] nodes;
        boolean isEnd;
        TrieNode() {
            nodes = new TrieNode[26];
            isEnd = false;
        }
    }

    TrieNode root;
    /** Initialize your data structure here. */
    public Trie() {
        root = new TrieNode();
    }

    /** Inserts a word into the trie. */
    public void insert(String word) {
        TrieNode node = root;
        for (char c : word.toCharArray()) {
            if (node.nodes[c - 'a'] == null) {
                node.nodes[c - 'a'] = new TrieNode();
            }
            node = node.nodes[c - 'a'];
        }
        node.isEnd = true;
    }

    /** Returns if the word is in the trie. */
    public boolean search(String word) {
        TrieNode node = helper(word);
        return node != null && node.isEnd;
    }

    /** Returns if there is any word in the trie that starts with the given prefix. */
    public boolean startsWith(String prefix) {
        return helper(prefix) != null;
    }

    private TrieNode helper(String word) {
        TrieNode node = root;
        for (char c : word.toCharArray()) {
            if (node.nodes[c - 'a'] == null) {
                return null;
            }
            node = node.nodes[c - 'a'];
        }
        return node;
    }
}

/**
 * Your Trie object will be instantiated and called as such:
```

```
* Trie obj = new Trie();
* obj.insert(word);
* boolean param_2 = obj.search(word);
* boolean param_3 = obj.startsWith(prefix);
*/
```