Our Solution(s) Run Code

Solution 1 Solution 2 Solution 3

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

#include <vector>
#include <unordered_map>
using namespace std;

class TrieNode {
public:
  unordered_map<char, TrieNode *> children;
  string word;
};

class Trie {
public:
  TrieNode *root;
  char endSymbol;

  Trie() {
    this->root = new TrieNode();
    this->endSymbol = '*';
  }

  void insert(string str) {
    TrieNode *current = this->root;
    for (int i = 0; i < str.length(); i++) {
      char letter = str[i];
      if (current->children.find(letter) == current->children.end()) {
        TrieNode *newNode = new TrieNode();
        current->children.insert({letter, newNode});
      }
      current = current->children[letter];
    }
    current->children.insert({this->endSymbol, NULL});
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
#include <vector>
using namespace std;

vector<bool> multiStringSearch(string bigString, vector<string> smallS
  // Write your code here.
  return {};
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.