```
#include <iostream>
#include <vector>
#include <string>
using namespace std;

// Function to find the maximum number of substrings that can be removed
int removeSubstrings(string mainString, const vector<string>& substrings) {
    int maxRemovals = 0;

    for (const string& sub : substrings) {
        size_t pos = mainString.find(sub);
        if (pos != string::npos) {
            string updatedString = mainString;
            updatedString.erase(pos, sub.length());

            maxRemovals = max(maxRemovals, 1 + removeSubstrings(updatedString, substrings));
        }
    }

    return maxRemovals;
}

int main() {
    int n;
    cin >> n;  // Input the number of substrings

    vector<string> substrings(n);
    for (int i = 0; i < n; i++) {
        cin >> substrings[i];  // Input substrings
    }

    string mainString;
    cin >> mainString;  // Input the main string

    // Calculate the result
    int result = removeSubstrings(mainString, substrings);

    // Print result without any extra spaces or newlines after the result
    cout << result;  // Do NOT use endl to avoid extra newline

    return 0;
}
```