Word Break in C++

```
#include <iostream>
#include <unordered_set>
#include <string>

using namespace std;

void wordBreak(string str, string ans,
unordered_set<string>& dict) {
    if (str.length() == 0) {
        cout << ans << endl;
        return;
    }

    for (int i = 0; i < str.length(); i++) {
        string left = str.substr(0, i + 1);
        if (dict.find(left) != dict.end()) {
            string right = str.substr(i + 1);
            wordBreak(right, ans + left + " ", dict);
        }
    }
}

int main() {
    int n = 5;
    unordered_set<string> dict = {"microsoft", "hiring",
"at", "kolkata"};
    string sentence = "microsofthiring";

    wordBreak(sentence, "", dict);

    return 0;
}
```

Step-by-Step Execution:

1. **First Call**: wordBreak("microsofthiring", "", dict)
    - **Current string**: "microsofthiring"
    - **Dictionary**: {"microsoft", "hiring", "at", "kolkata"}
    - **Answer so far**: ""
    - Loop through the string, checking for each substring that is present in the dictionary:
    - **i = 0**: Substring "m" is not in the dictionary.
    - **i = 1**: Substring "mi" is not in the dictionary.
    - **i = 2**: Substring "mic" is not in the dictionary.
    - **i = 3**: Substring "micro" is not in the dictionary.
    - **i = 4**: Substring "micr" is not in the dictionary.
    - **i = 5**: Substring "micro" is not in the dictionary.
    - **i = 6**: Substring "microsoft" **is** in the dictionary.
        - Now the string becomes "hiring", and the answer so far is "microsoft ".
    - **Recursively call**: wordBreak("hiring", "microsoft ", dict)

2. **Second Call**: wordBreak("hiring", "microsoft ", dict)
    - **Current string**: "hiring"
    - **Dictionary**: {"microsoft", "hiring", "at", "kolkata"}
    - **Answer so far**: "microsoft "
    - Loop through the string "hiring", checking for each substring:
    - **i = 0**: Substring "h" is not in the dictionary.
    - **i = 1**: Substring "hi" is not in the dictionary.
    - **i = 2**: Substring "hir" is not in the dictionary.
    - **i = 3**: Substring "hiri" is not in the dictionary.
    - **i = 4**: Substring "hiring" **is** in the dictionary.
        - Now the string becomes "" (empty), and the answer so far is "microsoft hiring ".
    - **Recursively call**: wordBreak("", "microsoft hiring ", dict)

	3. **Base Case**: wordBreak("", "microsoft hiring ", dict) ○ **Current string**: "" ○ **Answer so far**: "microsoft hiring " ○ Since the string is empty, print the answer: "microsoft hiring"
Output:- microsoft hiring	