1

```cpp
#include <iostream>
#include <vector>
#include <unordered_map>

using namespace std;

vector<int> twoSum(vector<int>& nums, int target) {
    unordered_map<int, int> num_map;
    for (int i = 0; i < nums.size(); ++i) {
        int complement = target - nums[i];
        if (num_map.find(complement) != num_map.end()) {
            return {num_map[complement], i};
        }
        num_map[nums[i]] = i;
    }
    return {};
}

int main() {
    vector<int> nums = {2, 7, 11, 15};
    int target = 9;
    vector<int> result = twoSum(nums, target);
    if (!result.empty()) {
        cout << "索引: " << result[0] << " & " << result[1] << endl;
    } else {
        cout << "未找到" << endl;
    }
    return 0;
}
```

2

```cpp
#include <iostream>
#include <vector>
#include <string>

using namespace std;

void computeNextArray(const string& pattern, vector<int>& next) {
    int m = pattern.length();
    next[0] = -1;
    int k = -1;
    for (int j = 1; j < m; ++j) {
        while (k >= 0 && pattern[k + 1] != pattern[j]) {
            k = next[k];
        }
        if (pattern[k + 1] == pattern[j]) {
            ++k;
        }
        next[j] = k;
    }
}

int KMP(const string& text, const string& pattern) {
    int n = text.length();
    int m = pattern.length();
    vector<int> next(m);
    computeNextArray(pattern, next);

    int j = 0;
    for (int i = 0; i < n; ++i) {
        while (j >= 0 && text[i] != pattern[j]) {
            j = next[j];
        }
        if (text[i] == pattern[j]) {
            ++j;
        }
        if (j == m) {
            return i - m + 1; // 返回匹配的起始位置
        }
    }
```

```
    return -1; // 未找到匹配
}
```