**SOLUTIONS IN C++**

1.

#include <iostream>

#include <string>

using namespace std;

int lengthOfLastWord(string s) {

int length = 0;

int i = s.length() - 1;

while (i >= 0 && s[i] == ' ') {

i--;

}

while (i >= 0 && s[i] != ' ') {

length++;

i--;

}

return length;

}

int main() {

string s1 = "Hello World";

string s2 = " fly me to the moon ";

string s3 ="my mom is great";

cout << "Length of the last word in s1: " << lengthOfLastWord(s1) << endl;

cout << "Length of the last word in s2: " << lengthOfLastWord(s2) << endl;

cout<<"Length of last word in s3:"<<lengthOfLastWord(s3)<<endl;

return 0;

}

2.

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

vector<vector<int>> fourSum(vector<int>& nums, int target) {

vector<vector<int>> result;

int n = nums.size();

if (n < 4) {

return result;

}

sort(nums.begin(), nums.end());

for (int i = 0; i < n - 3; i++) {

if (i > 0 && nums[i] == nums[i - 1]) {

continue;

}

for (int j = i + 1; j < n - 2; j++) {

if (j > i + 1 && nums[j] == nums[j - 1]) {

continue;

}

int left = j + 1;

int right = n - 1;

while (left < right) {

int sum = nums[i] + nums[j] + nums[left] + nums[right];

if (sum == target) {

result.push\_back({nums[i], nums[j], nums[left], nums[right]});

while (left < right && nums[left] == nums[left + 1]) left++;

while (left < right && nums[right] == nums[right - 1]) right--;

left++;

right--;

} else if (sum < target) {

left++;

} else {

right--;

}

}

}

}

return result;

}

int main() {

vector<int> nums1 = {1, 0, -1, 0, -2, 2};

int target1 = 0;

vector<vector<int>> result1 = fourSum(nums1, target1);

vector<int> nums2 = {2, 2, 2, 2, 2};

int target2 = 8;

vector<vector<int>> result2 = fourSum(nums2, target2);

for (vector<int> quad : result1) {

cout << "[ ";

for (int num : quad) {

cout << num << " ";

}

cout << "]" << endl;

}

for (vector<int> quad : result2) {

cout << "[ ";

for (int num : quad) {

cout << num << " ";

}

cout << "]" << endl;

}

return 0;

}