

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное
учреждение высшего образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Инженерная школа информационных технологий и робототехники
Отделение информационных технологий
Направление информатика и вычислительная техника

Отчет
по лабораторной работе №3

по дисциплине
«ПРОГРАММИРОВАНИЕ»

Работа с базовым контейнером string на языке программирования C++

Выполнил:

Студент группы 8B41

Ван Чжиюань

Проверил:

Ассистент ОИТ ИШИТР

А.В. Кузнецов

Томск 2025

Цель работы

Получить навыки работы с базовым контейнером string на языке программирования C++..

Задание

Выполнить по одному заданию из каждого блока, которые представлены ниже. Проект для выполнения заданий собрать и отладить на языке программирования C++ с применением CMake в VSCode.

Блок 1

1. Даны string s и vector words, где каждый элемент вектора- строка, состоящая из строчных латинских букв. Все строки- только из строчных латинских букв. Написать функцию, которая возвращает true, если s является акронимом для words. Это происходит в том случае, если мы соединяем первый символ каждого элемента вектора words по порядку и получаем строку s. Например, строка "тпу" может быть сформирована из вектора ["томский", "политехнический", "университет"], но не из вектора ["томский", "университет", "политехнический"].

На рисунке 1 мы начинаем отладку функции isAcronym. Мы видим, что программа останавливается на `cin >> abbr;`, интерфейс отладки отображает приглашение на ввод «acronym:», а abbr и words в окне переменных не инициализируются. На этом этапе программа ожидает ввода аббревиатуры.

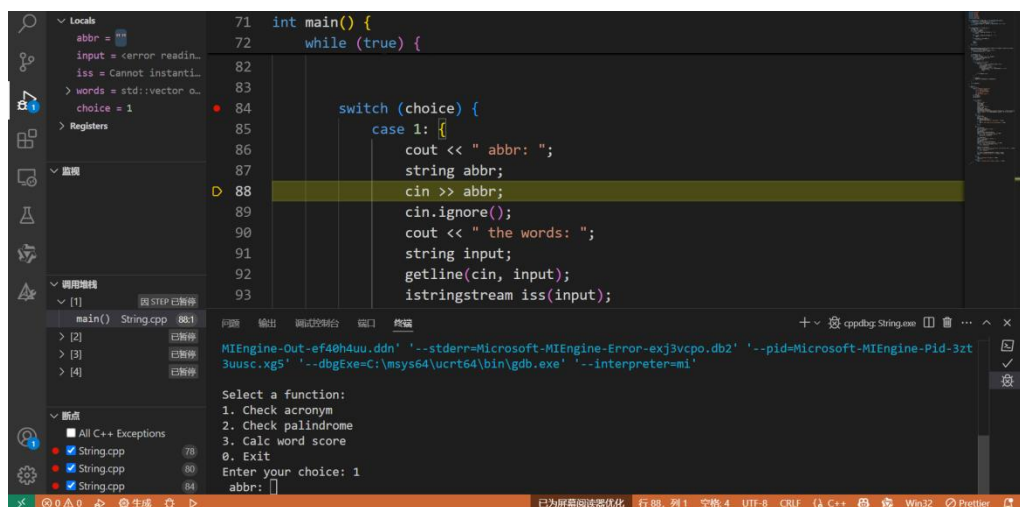


Рисунок 1 – Результат отладки функции isAcronym

На рисунке 2 введены аббревиатура и слово. В окне переменных abbr — это «tpu», а input — это введенная строка слов. После завершения ввода программа разделит входную строку на векторы слов, а затем вызовет функцию isAcronym для оценки. »

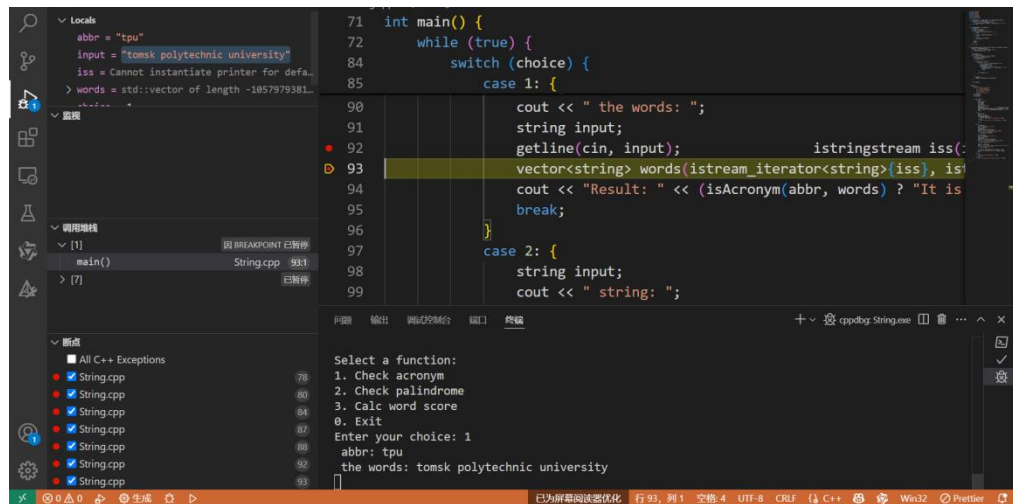


Рисунок 2 – Результат отладки функции isAcronym

На рисунке 3 мы видим консоль отладки, отображающую «Result: 1». Значит, что аббревиатура «tpu» соответствует входной фразе. Это показывает, что функция isAcronym правильно определила, что «tpu» является аббревиатурой входной фразы. »

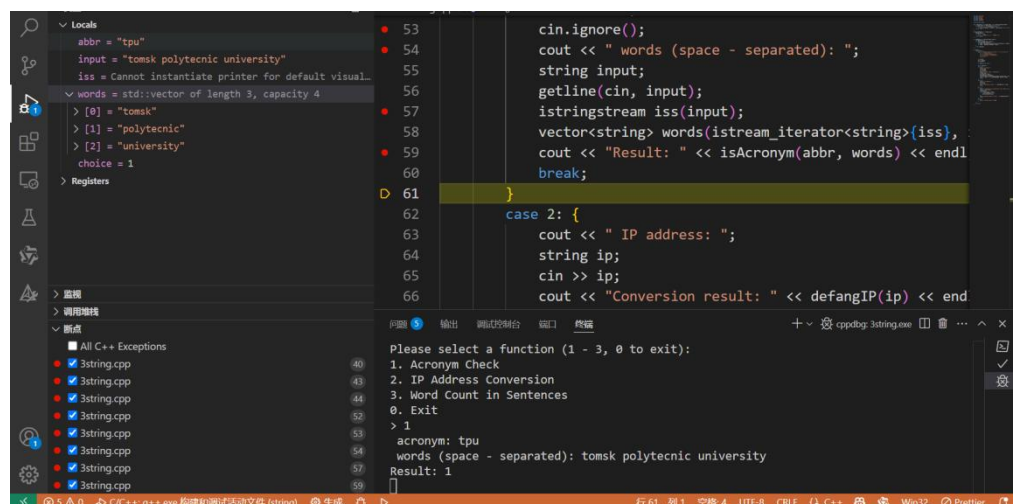


Рисунок 3 – Результат отладки функции isAcronym

Блок 2

1. Написать функцию, проверяющую, что вводимая пользователем строка, которая состоит из строчных латинских букв и пробелов, является палиндромом (читается вперёд и назад одинаково). Строка может содержать произвольное количество пробелов подряд; таким образом, для "never odd or even" программа должна вернуть true.

На рисунках 4, 5, 6 и 7 показан процесс проверки палиндрома с вводом «никогда нечетное и нечетное». Left=0, right=16.

Мы видим, что левое и правое постепенно приближаются к середине, и в конечном итоге левое и правое встретятся (левое == правое). В этот момент условие внешнего цикла while left < right не выполняется, и внешний цикл while прерывается.

Поскольку каждая пара соответствующих символов равна в течение всего процесса сравнения, функция возвращает значение true, указывая на то, что входная строка «никогда не бывает четной или нечетной» является палиндромом. Рисунок 8 показывает результаты.

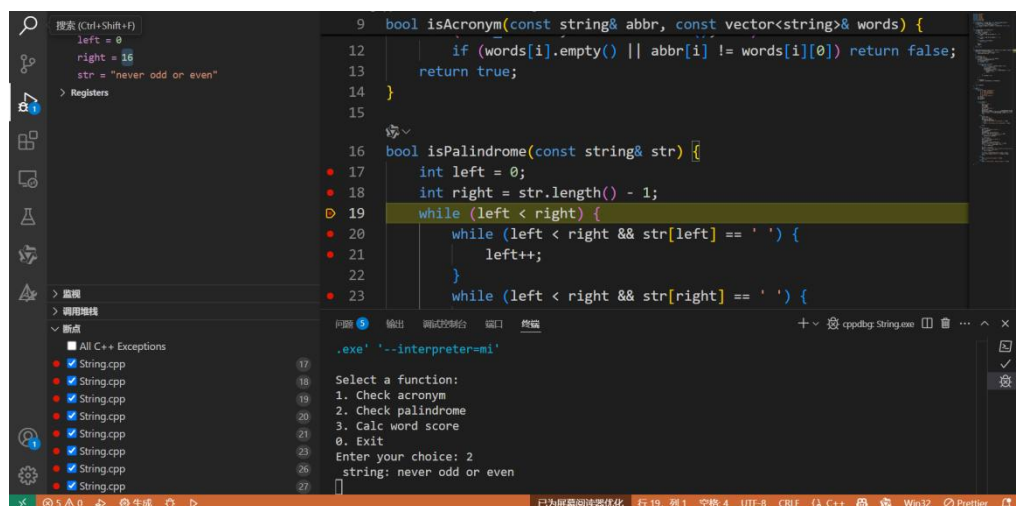


Рисунок 4 – Результаты отладки функции `isPalindromе`

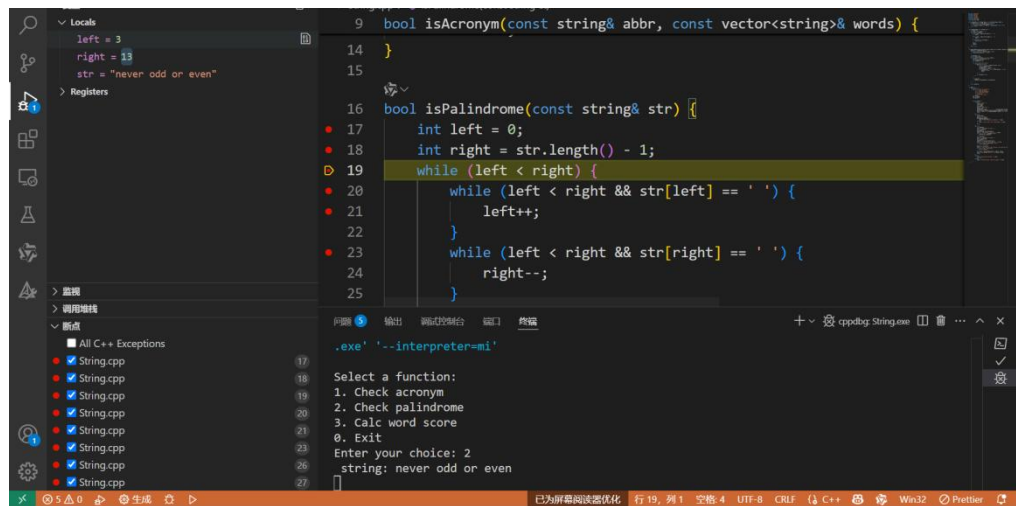


Рисунок 5 – Результаты отладки функции isPalindrome

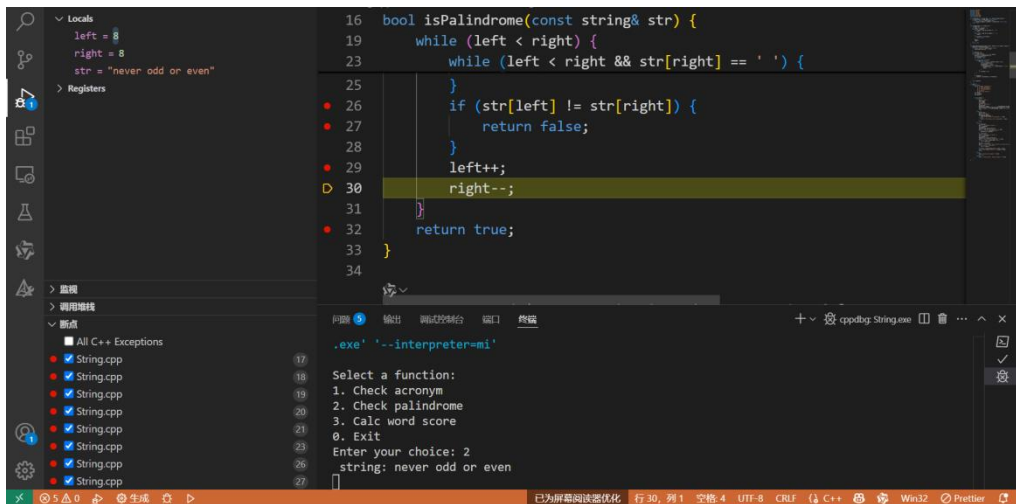


Рисунок 6 – Результаты отладки функции isPalindrome

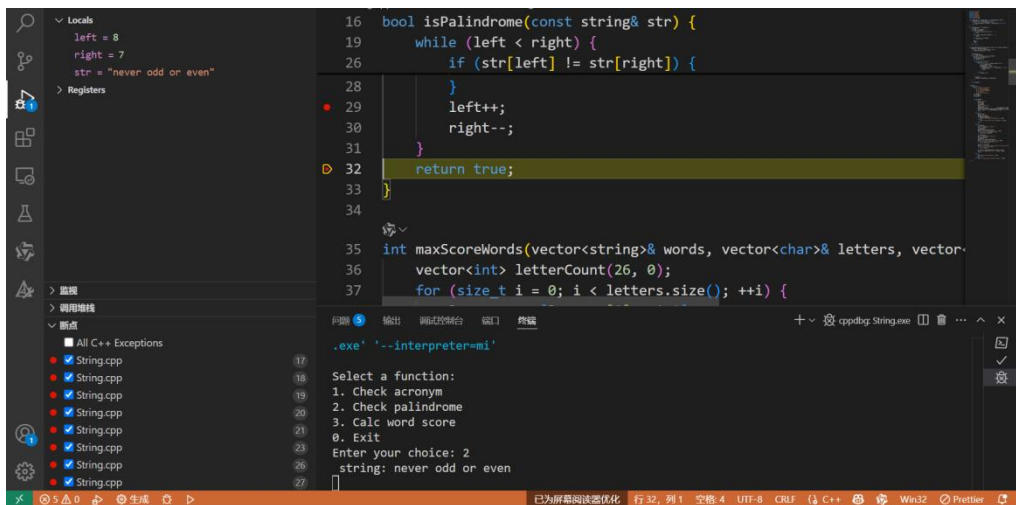


Рисунок 7 – Результаты отладки функции isPalindrome

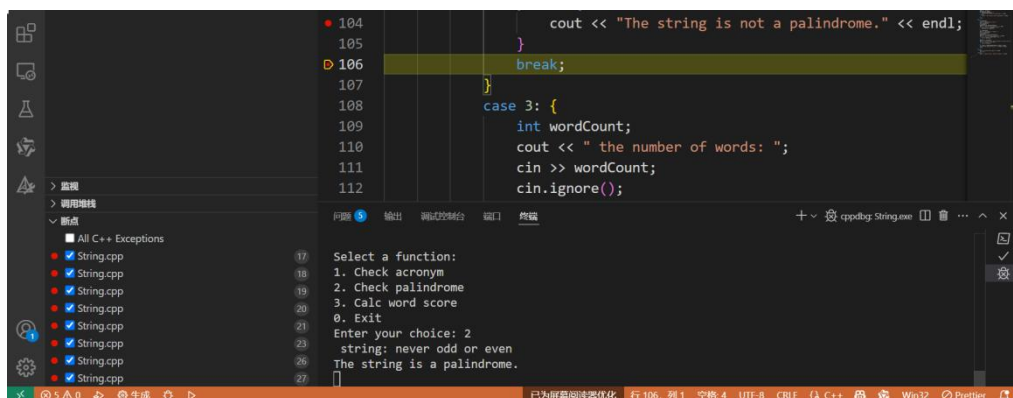


Рисунок 8 – Результат отладки функции defangIP

На рисунках 9 и 10 показаны изменения переменных letterCount, maxScore, n, mask, currentCount, currentScore и isValid. Результаты показаны на рисунке 11.

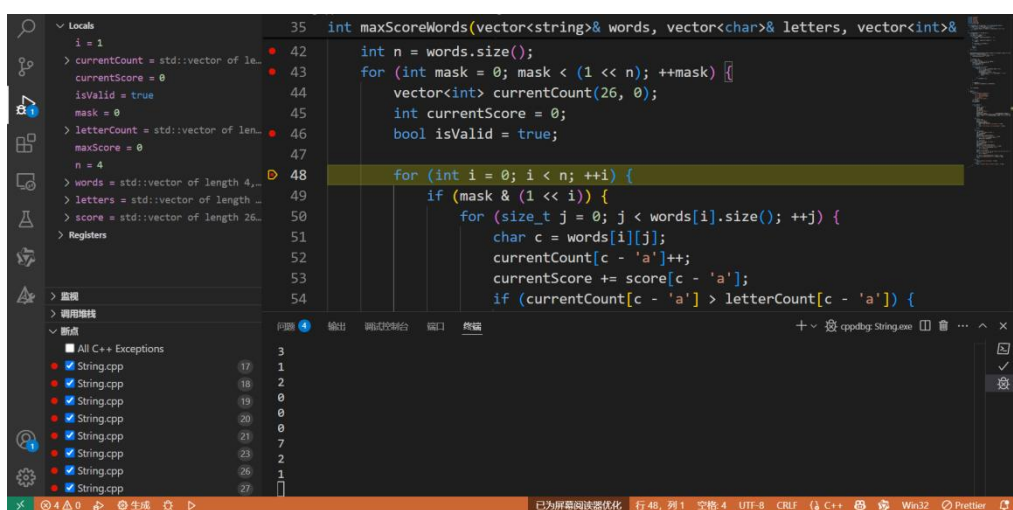


Рисунок 9 – Результат отладки функции maxScoreWords

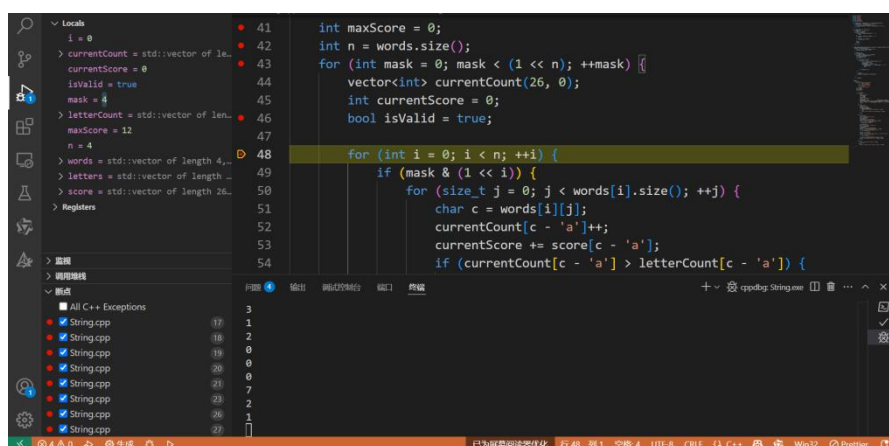


Рисунок 10 – Результат отладки функции maxScoreWords



Рисунок 11 – Результат отладки функции maxScoreWords

На рисунке 12,13 показан результат выполнения исполняемого файла, созданного stake.

```
Select a function:
1. Check acronym
2. Check palindrome
3. Calc word score
0. Exit
Enter your choice: 1
abbr: tpu
the words: tomsk polytechnic university
Result: It is an acronym.

Select a function:
1. Check acronym
2. Check palindrome
3. Calc word score
0. Exit
Enter your choice: 2
string: abbcca
The string is not a palindrome.

Select a function:
1. Check acronym
2. Check palindrome
3. Calc word score
0. Exit
Enter your choice: 3
the number of words: 4
each word on a new line:
dog
cat
dad
mom
the number of letters: 9
each letter on a new line:
a
```

Рисунок 12 – Результаты в исполняемом файле


```
b
b
c
d
d
d
g
o
the scores for the 26 letters, one score per line:
1
0
2
3
0
0
0
0
9
7
6
5
4
3
2
1
0
0
0
0
0
4
5
6
3
4
The maximum score is: 7
```

Рисунок 13 – Результаты в исполняемом файле

Выводы

В данной работе путем написания функции `isAcronym` реализована функция определения, является ли заданная аббревиатура аббревиатурой набора списков слов. Используются эффективно строковые и векторные структуры данных для логических рассуждений. Использован тип `size_t` для обеспечения безопасности индексов и размеров, связанных с контейнером, а также обработаны граничные условия (например, пустые слова) во время обхода. Написанной функцией `isPalindrome` проверена, является ли входная строка палиндромом. Написанной функцией `maxScoreWords` вычислена максимальная оценка слова, состоящего из заданного набора букв. Исследованы сложные методы разработки алгоритмов, такие как использование битовых масок для решения комбинационных задач и оптимизация обработки крупномасштабных комбинационных ситуаций.

Приложение 1 – Код программы

По [ссылке](#), ведущий на репозиторий в codelab.tpu.ru, представлены файлы исходного кода проекта к данной лабораторной работе.

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

bool isAcronym(const string& s, const vector<string>& words) {
    if (s.size() != words.size()) return false;
    for (size_t i = 0; i < words.size(); i++)
        if (words[i].empty() || s[i] != words[i][0]) return false;
    return true;
}

bool isPalindrome(const string& str) {
    int left = 0;
    int right = str.length() - 1;
    while (left < right) {
        while (left < right && str[left] == ' ') {
            left++;
        }
        while (left < right && str[right] == ' ') {
            right--;
        }
        if (str[left] != str[right]) {
            return false;
        }
        left++;
        right--;
    }
    return true;
}

int maxScoreWords(vector<string>& words, vector<char>& letters, vector<int>& score)
{
    vector<int> lettersCount(26, 0);
    for (size_t i = 0; i < letters.size(); i++) {
        lettersCount[letters[i] - 'a']++;
    }

    int maxScore = 0;
    int n = words.size();
    for (int mask = 0; mask < (1 << n); mask++) {
        vector<int> currentCount(26, 0);
        int currentScore = 0;
        bool isValid = true;

        for (int i = 0; i < n; i++) {
            if (mask & (1 << i)) {
                for (size_t j = 0; j < words[i].size(); j++) {
                    char c = words[i][j];
                    currentCount[c - 'a']++;
                    currentScore += score[c - 'a'];
                }

                if (currentCount[c - 'a'] > lettersCount[c - 'a']) {
```

```

        isValid = false;
        break;
    }
    }
    if (!isValid) break;
}
}
if (isValid) maxScore = max(maxScore, currentScore);
}
return maxScore;
}

int main() {
    while (true) {
        cout << "\nSelect a function:\n"
            << "1. Check acronym\n"
            << "2. Check palindrome\n"
            << "3. Calc word score\n"
            << "0. Exit\n"
            << "Enter your choice: ";
        int choice;
        cin >> choice;
        cin.ignore();

        switch (choice) {
            case 1: {
                cout << " abbr: ";
                string abbr;
                cin >> abbr;
                cin.ignore();
                cout << " the words: ";
                string input;
                getline(cin, input);
                istringstream iss(input);
                vector<string> words(istream_iterator<string>{iss},
istream_iterator<string>());
                cout << "Result: " << (isAcronym(abbr, words) ? "It is an acronym." : "It
is not an acronym.") << endl;
                break;
            }
            case 2: {
                string input;
                cout << " string: ";
                getline(cin, input);
                if (isPalindrome(input)) {
                    cout << "The string is a palindrome." << endl;
                }
                else {
                    cout << "The string is not a palindrome." << endl;
                }
                break;
            }
            case 3: {
                int wordCount;
                cout << " the number of words: ";
                cin >> wordCount;
                cin.ignore();
                vector<string> words(wordCount);
                cout << " each word on a new line: " << endl;
                for (int i = 0; i < wordCount; ++i) {
                    getline(cin, words[i]);
                }
                int letterCount;
                cout << " the number of letters: ";
            }
        }
    }
}

```

```

    cin >> letterCount;
    cin.ignore();
    vector<char> letters(letterCount);
    cout << " each letter on a new line: " << endl;
    for (int i = 0; i < letterCount; ++i) {
        cin >> letters[i];
    }
    vector<int> score(26);
    cout << " the scores for the 26 letters, one score per line: " << endl;
    for (int i = 0; i < 26; ++i) {
        cin >> score[i];
    }
    int result = maxScoreWords(words, letters, score);
    cout << "The maximum score is: " << result << endl;
    break;
}
case 0:
    cout << "Exiting the program." << endl;
    return 0;
default:
    cout << "Invalid choice. Please try again." << endl;
}
}
}

```