|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3** | |
| **по дисциплине** | |
| **«**Структуры и алгоритмы обработки данных**»**  **Тема: «Структура данных – строка»** | |
|  | |
| Выполнил студент группы ИКБО-09-22 | Гришин А. В. |
| Принял преподаватель | Филатов А.С. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2023

# **Цель работы**

Получить навыки в разработке алгоритмов обработки текста – извлечение отдельных элементов, а также в использовании средств языка Си и С++ для реализации алгоритмов обработки текстовых данных.

1. **Постановка задачи**
2. Разработать программу согласно задаче варианта, используя для представления обрабатываемого в программе текста нуль терминальную строку и средства языка С для выполнения операций над этой строкой.
3. Разработать программу согласно задаче варианта, используя для представления обрабатываемого в программе текста строку string стандартной библиотеки шаблонов и возможности класса для выполнения действий со строкой.
4. Составить отчет, отобразив в нем описание выполнения всех этапов разработки, тестирования и код всей программы со скриншотами результатов тестирования.

Персональный вариант – 7.

Таблица 1. Задание варианта

|  |
| --- |
| Дано предложение, состоящее из слов, разделенных запятой или пробелами. Вывести слова предложения предварительно преобразовав их следующим образом:  - перенести последнюю букву в начало слова;  - удалить из слова повторные вхождения каждой буквы. |

# **Решение**

* 1. **Краткая теория по теме**

Текст — это последовательность символов. Строка в Си – это массив символов кодировки ASCII расширенная, заканчивающийся символом \0 (null - символ конца строки).

В языке программирования C++ строку можно задать двумя способами – в виде статического массива типа char (строковая переменная) и с помощью шаблонной строки класса string. Строковые переменные объявляются как:

|  |
| --- |
| char str[10] = ”12345”; // Выделение памяти под 10 элементов, из них 1 под \0. |

При инициализации строковой переменной значение вписывается в выделенную память в направлении слева направо. Так в ps1 значение 1 займет первую ячейку строки с индексом 0, а далее по порядку остальные. Поэтому объем памяти для переменной строкового типа равен размеру массива, но количество символов в строке на 1 меньше. Также строки такого типа можно объявить, как указатель или динамический массив типа char.

Ввод-вывод строк в языке программирования C++ в основном выполняется с помощью специальных потоков ввода cin и вывода cout библиотеки <iostream>.

|  |
| --- |
| cin >> строковая\_переменная;  cout << строковая\_переменная; |

Однако иногда удобно использовать специальные функции (стандартного потока cin), считывающие, например, всю строку, введенную пользователем. Так функция getline() считывает всю строку от пользователя до \n (переход на новую строку). Также существуют и другие функции, облегчающие работу с вводом текста, например вспомогательные функции для нахождения длины строки, символа в строке, перевода строки в число и т.п.

Если же говорить о шаблонной строке класса string, то сам класс string определен в стандартной библиотеке шаблонов STL. Для его использования необходимо подключить заголовочный файл <string>. В случае с string, мы тоже можем использовать вспомогательные функции для замены символов в строке, поиска индекса определенного символа и другие функции для работы со строкой.

* 1. **. Дополнительные функции**

Для упрощения работы над реализацией программы и повышения читаемости кода введем дополнительные функции, а именно для вывода массива слов, полученного после всей работы программы (отдельно для массива char и вектора).

|  |
| --- |
| void WordsOutput(char\*\* words) {  for (int i = 0; i < 100; i++) {  if (words[i] == NULL) break;  cout << words[i] << " ";  }  cout << endl;  }  void WordsOutput(vector<string> words) {  for (const auto& t : words) {  cout << t << " ";  }  cout << endl;  } |

* 1. **. Выполнение задания, используя для представления обрабатываемого в программе текста нуль терминальную строку и средства языка С для выполнения операций над этой строкой.**

Для работы со строковой переменной – массивом символов (в нашем случае букв), необходимо создать сам массив типа char, за размерность которого возьмём начальное значение 100.

Далее идёт считывание всей строки с помощью gets\_s() и вызов функции CharEditingWords(), в качестве аргумента которая принимает введенную пользователем строку, передаваемую по ссылке.

В самой функции инициализируются переменная для запоминания количества слов в строке и массив char для записи этих слов.

Далее, мы проходимся по строке и добавляем каждое отдельное слово, разделённое запятой или пробелом. При этом мы считаем количество слов, содержащихся в строке, и заносим это в переменную wordscount.

После этого, в течение прохода по массиву, программа для каждого отдельного элемента производит две отдельные операции: перестановку последней буквы каждого элемента в его начало посредством сдвига символов внутри элемента и удаление повторных вхождений каждой буквы в каждый отдельный элемент на основе сравнения символов внутри него.

|  |
| --- |
| // Редактирование null терминальной строки  void CharEditingWords(auto& sentence)  {  char\* words[100];  char razd[] = " ,";  int wordscount = 0;  words[0] = strtok(sentence, razd);  while (words[wordscount] != NULL) {  words[++wordscount] = strtok(NULL, razd);  }  for (int j = 0; j < wordscount; j++) {  auto curword = words[j];  int len = strlen(curword);  char temp = curword[len - 1];  for (int l = len - 1; l > 0; l--) {  curword[l] = curword[l - 1];  }  curword[0] = temp;  words[j] = curword;  int r = 0, len2 = len;  for (int l = len - 1; l > 0; l--) {  for (int k = l - 1; k >= 0; k--) {  if (curword[l] == curword[k]) {  for (int m = k; m < len - 1; m++) {  curword[m] = curword[m + 1];  }  len--;  r++;  }  }  }  curword[len2 - r] = '\0';  words[j] = curword;  }  WordsOutput(words);  } |

По завершении всех операций в конце функции вызывается ещё одна — для вывода всех элементов массива words.

* 1. **Выполнение задания, используя для представления обрабатываемого в программе текста строку string стандартной библиотеки шаблонов и возможности класса для выполнения действий со строкой.**

В данной реализации задания объявляется строка типа string и для неё считывается значение с клавиатуры.

Далее, в функции StringEditingWords() инициализируется вектор words для записи слов из строки. После этого, в течение прохода по вектора, программа для каждого отдельного элемента производит две отдельные операции: перестановку последней буквы каждого элемента в его начало посредством изменения размера вектора и вставки по индексу и удаление повторных вхождений каждой буквы в каждый отдельный элемент на основе сравнения символов внутри него.

|  |
| --- |
| // Редактирование строки типа string  void StringEditingWords(string& sentence) {  sentence.erase(remove\_if(sentence.begin(), sentence.end(), ::ispunct), sentence.end());  stringstream ss(sentence);  string token;  vector<string> words;  while (ss >> token) {  words.push\_back(token);  }  for (auto& word : words) {  if (!word.empty()) {  char lastChar = word.back();  word.pop\_back();  word.insert(0, 1, lastChar);  string uniqueChars;  for (char c : word) {  if (uniqueChars.find(c) == string::npos) {  uniqueChars += c;  }  }  word = uniqueChars;  }  }  WordsOutput(words);  } |

По завершении всех операций в конце функции вызывается ещё одна — для вывода всех элементов вектора words.

* 1. **Описание работы программы и её интерфейса**

В самом начале программы выводится меню выбора способа реализации задания, с помощью нуль терминальной строки или строки string стандартной библиотеки шаблонов.

При выборе первого пункта меню, программа входит в бесконечный цикл взаимодействия. Далее требуется ввести строку, которая в ходе работы программы будет преобразована в массив слов, обработанных в соответствии с условием задания (рис. 1).

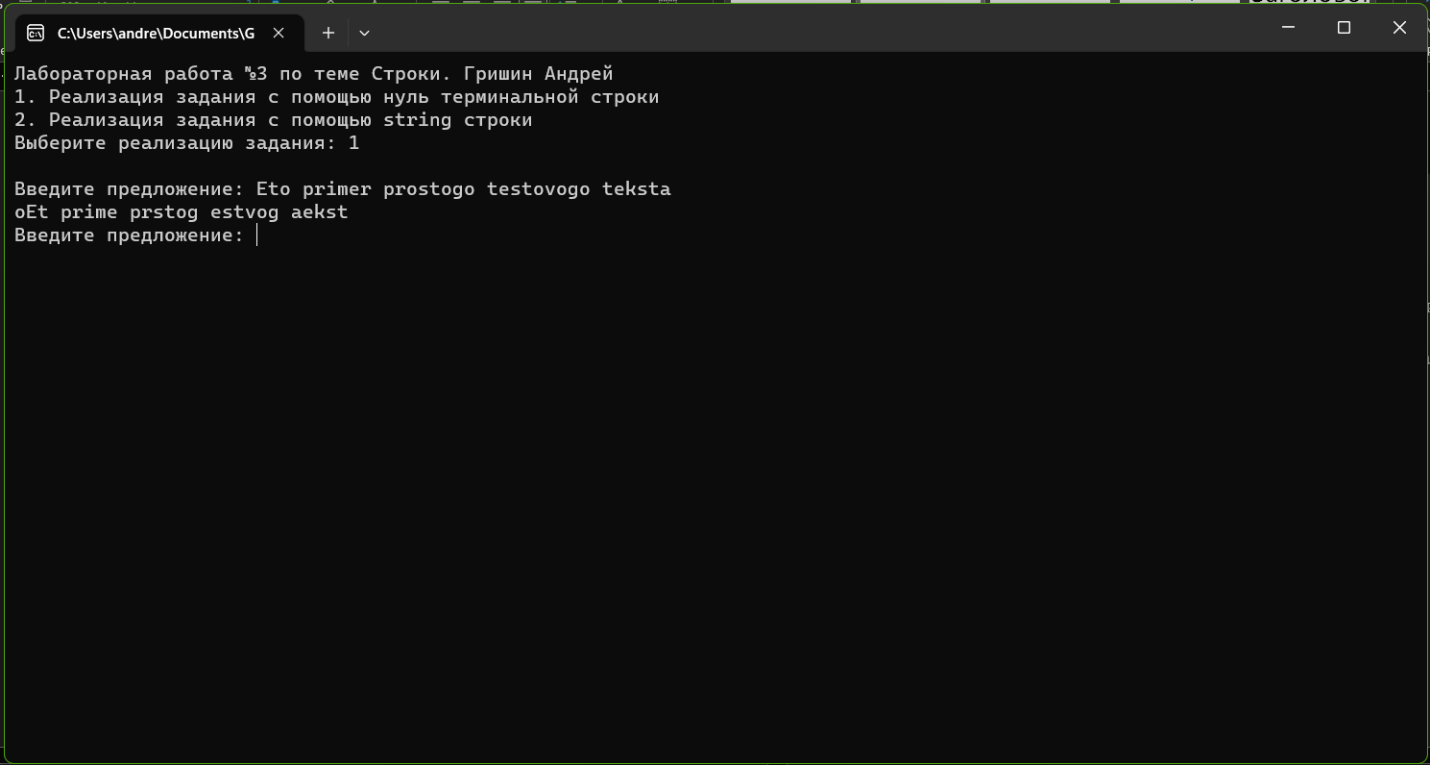


Рисунок 1. Интерфейс работы программы

с нуль терминальной строкой

Для строки string всё происходит точно также, отличается лишь реализация функций, описанных выше. Так программа также запрашивает строку на вход, преобразовывает в обработанный функцией вектор слов, а затем его выводит (рис. 2).

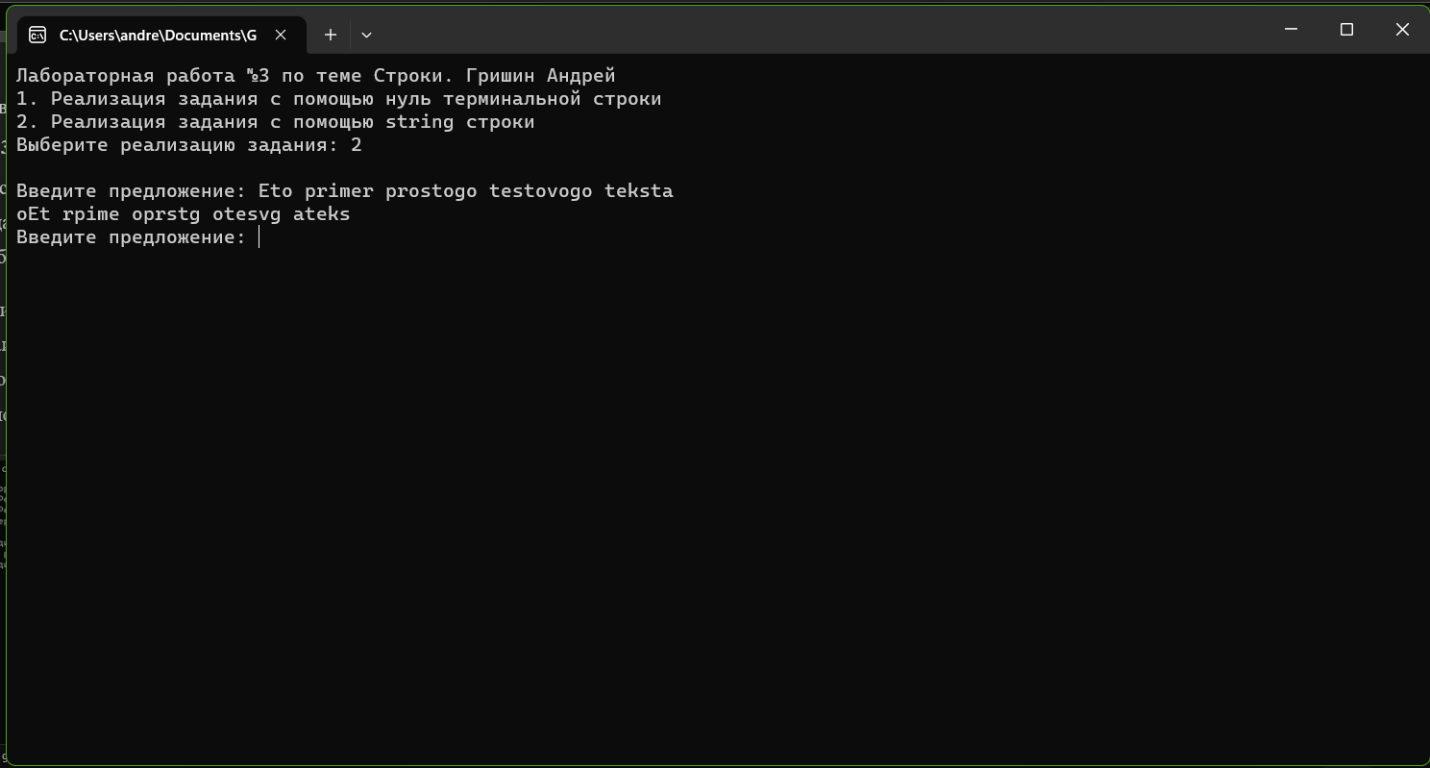


Рисунок 2. Интерфейс работы программы

с строкой типа string

1. **Тестирование**

Протестируем работу программы для всех способов её реализации.

Рассмотрим выполнение задания с помощью нуль терминальной строки (рис. 3). На выходе мы должны получить слова, обработанные по двум условиям: последняя буква слова должна быть перенесена на первую позицию и затем должны быть удалены из слова повторные вхождения каждой буквы (остаётся лишь самая последняя из всех повторяющихся букв), если такие есть.

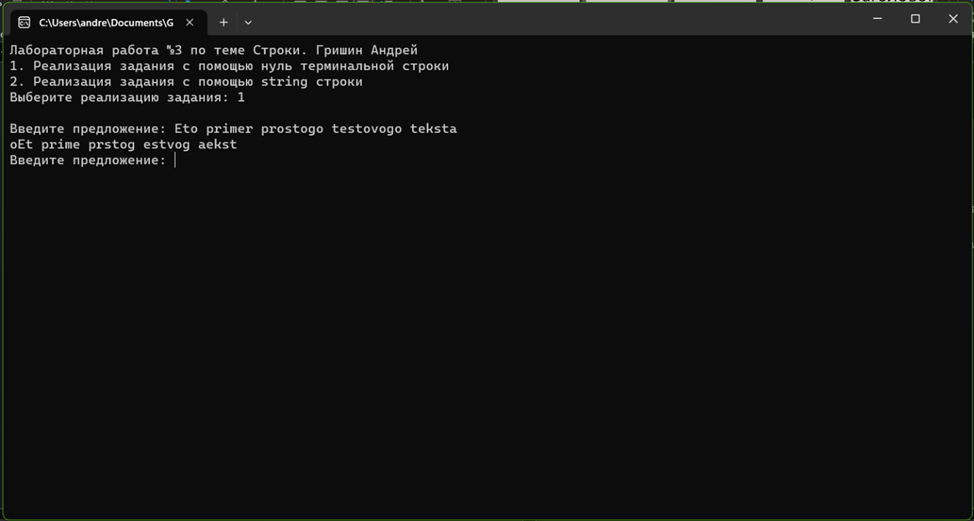


Рисунок 3. Тестирование задания с нуль терминальной строкой

Тестирование задания с помощью нуль терминальной строки завершено успешно. Протестируем реализацию задания для строки string. На выходе должны получить слова, обработанные по двум условиям: последняя буква слова должна быть перенесена на первую позицию и затем должны быть удалены из слова повторные вхождения каждой буквы (остаётся лишь первая из всех повторяющихся букв), если такие есть. (рис. 4)

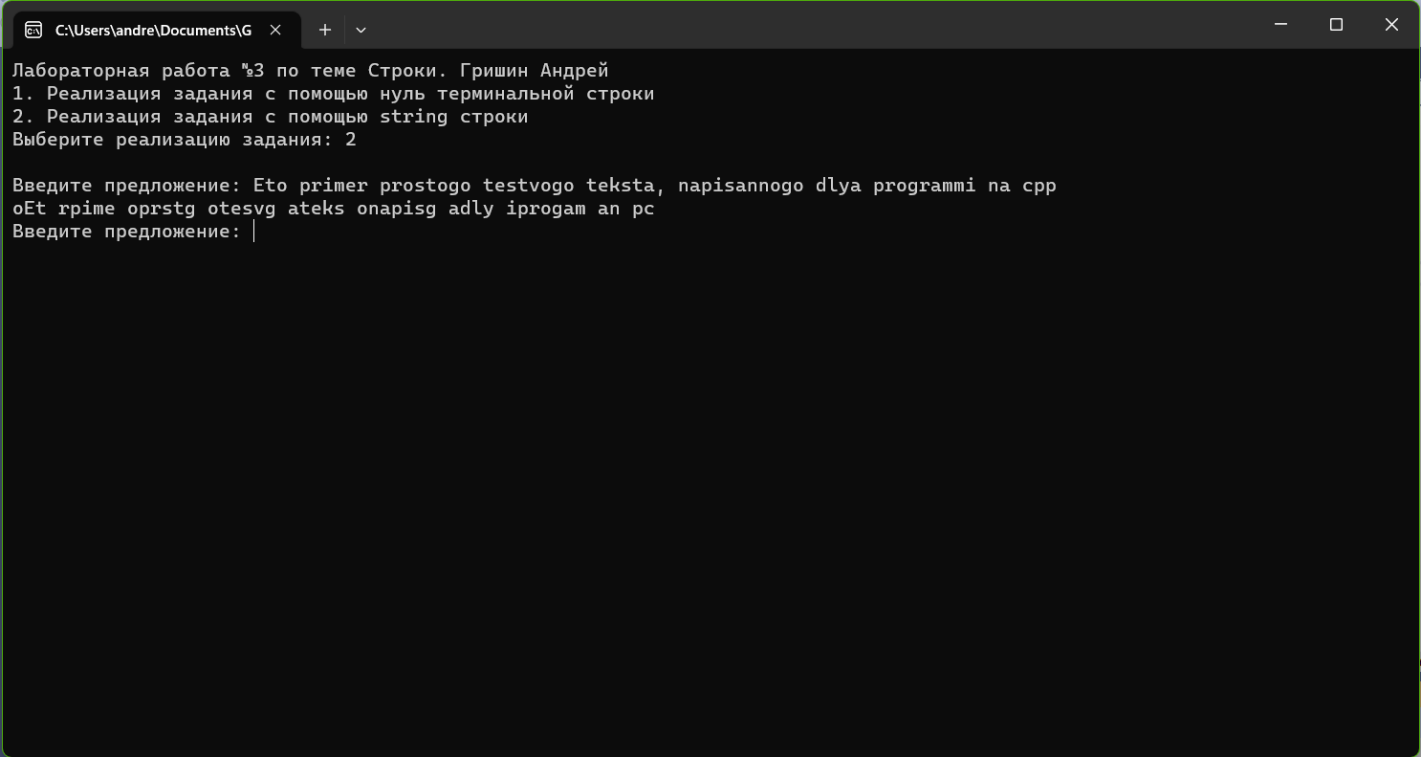


Рисунок 4. Тестирование задания со строкой string

Тестирование задания с помощью строки string завершено успешно.

1. **Вывод**

В результате выполнения работы я приобрел навыки по использованию для представления обрабатываемого в программе текста нуль терминальную строку и строку string стандартной библиотеки шаблонов, включая возможности класса, в языке программирования C++.

1. **Исходный код программы**

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include <iostream>  #include <string>  #include <vector>  #include <algorithm>  #include <sstream>  using namespace std;  void WordsOutput(char\*\* words) {  for (int i = 0; i < 100; i++) {  if (words[i] == NULL) break;  cout << words[i] << " ";  }  cout << endl;  }  void WordsOutput(vector<string> words) {  for (const auto& t : words) {  cout << t << " ";  }  cout << endl;  }  // Редактирование null терминальной строки  void CharEditingWords(auto& sentence)  {  char\* words[100];  char razd[] = " ,";  int wordscount = 0;  words[0] = strtok(sentence, razd);  while (words[wordscount] != NULL) {  words[++wordscount] = strtok(NULL, razd);  }  for (int j = 0; j < wordscount; j++) {  auto curword = words[j];  int len = strlen(curword);  char temp = curword[len - 1];  for (int l = len - 1; l > 0; l--) {  curword[l] = curword[l - 1];  }  curword[0] = temp;  words[j] = curword;  int r = 0, len2 = len;  for (int l = len - 1; l > 0; l--) {  for (int k = l - 1; k >= 0; k--) {  if (curword[l] == curword[k]) {  for (int m = k; m < len - 1; m++) {  curword[m] = curword[m + 1];  }  len--;  r++;  }  }  }  curword[len2 - r] = '\0';  words[j] = curword;  }  WordsOutput(words);  }  // Редактирование строки типа string  void StringEditingWords(string& sentence) {  sentence.erase(remove\_if(sentence.begin(), sentence.end(), ::ispunct), sentence.end());  stringstream ss(sentence);  string token;  vector<string> words;  while (ss >> token) {  words.push\_back(token);  }  for (auto& word : words) {  if (!word.empty()) {  char lastChar = word.back();  word.pop\_back();  word.insert(0, 1, lastChar);  string uniqueChars;  for (char c : word) {  if (uniqueChars.find(c) == string::npos) {  uniqueChars += c;  }  }  word = uniqueChars;  }  }  WordsOutput(words);  }  int vibor1() {  int menuChoice;  cout << "1. Реализация задания с помощью нуль терминальной строки" << endl;  cout << "2. Реализация задания с помощью string строки" << endl;  cout << "Выберите реализацию задания: "; cin >> menuChoice;  cout << endl;  return menuChoice;  }  int main()  {  setlocale(LC\_ALL, "Russian");  int menuChoice;  cout << "Лабораторная работа №3 по теме Строки. Гришин Андрей" << endl;  int num = vibor1();  cin.ignore();  switch (num)  {  case 1:  {  while (true)  {  cout << "Введите предложение: ";  char sentence[100];  gets\_s(sentence);  CharEditingWords(sentence);  }  break;  }  case 2:  {  while (true)  {  cout << "Введите предложение: ";  string sentence;  getline(cin, sentence);    StringEditingWords(sentence);  }  break;  }  }  } |