Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа

По дисциплине «Основы алгоритмизации и программирования»

На тему «Обработка символьной информации»

Выполнил:

Студент 1 курса 7 группы

Ленкевич Павел Андреевич

Преподаватель: асс. Андронова М.В.

2023, Минск

6. Выполнить задания из таблицы ниже, используя ***индексы*** для доступа к элементам массивов в первой программе и ***указатели*** − во второй программе. При написании программне использовать стандартные функции для строк символов.

**Вариант 9**

1. Написать программу, реализующую вставку подстроки **St** длиной **n1** в строку **S** с позиции номер **n2**.
2. Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков «+» и «–». Найти количество цифр.

**Задание 1.**

#include <iostream> //Подключение библиотеки

using namespace std; // Позволяет не писать std перед потоковыми операторами Ввода-вывода

int main() // Тело программы

{

setlocale(LC\_CTYPE, "Russian"); // Подключает русский язык

const int n1 = 100; // Максимальный размер строки

char St[n1], S[n1]; // Два массива для двух строк

cout << "Введите строку: "; cin.getline(St, n1); // Считывает размер строки и вставляет его в массив

int StIndex = 0, SIndex = 0, n; // Подключаем индексы для строк

cout << "Введите номер позиции: "; cin >> n;

while (St[StIndex] != '\0') { // Пока мы не дойдём до последнего символа, цикл будет повторятся

S[SIndex] = St[StIndex + n]; // Копируем символы во второй массив

SIndex++;

StIndex++;

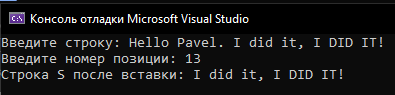
}

S[SIndex] = '\0'; // Добавляем завершающий символ во второй массив

cout << "Строка S после вставки: " << S << endl;

return 0;

}



**Задание 2.**

#include <iostream> //Подключение библиотеки

using namespace std; // Позволяет не писать std перед потоковыми операторами Ввода-вывода

int main() // Тело программы

{

setlocale(LC\_CTYPE, "Russian"); // Подключает русский язык

int i, len, maxlen; // Задаём переменные и их значения

const char\* str = "100111000011111000000"; // Вводим строку из 0 и 1, которая нам дана

for (i = 1, len = maxlen = 1; str[i]; i++) // Цикл на количество повторов

if (str[i] == str[i - 1]) { // Если элемент массива равен прошлому элементу массива...

len++; // то len увеличивается на 1 и

maxlen = len > maxlen ? len : maxlen; // Выполняется тернарный логический оператор

}

else // Если не удовлетворяет условию

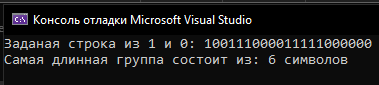
len = 1;

cout << "Заданая строка из 1 и 0: " << str << endl;

cout << "Самая длинная группа состоит из: " << maxlen << " символов" << endl;

return 0;

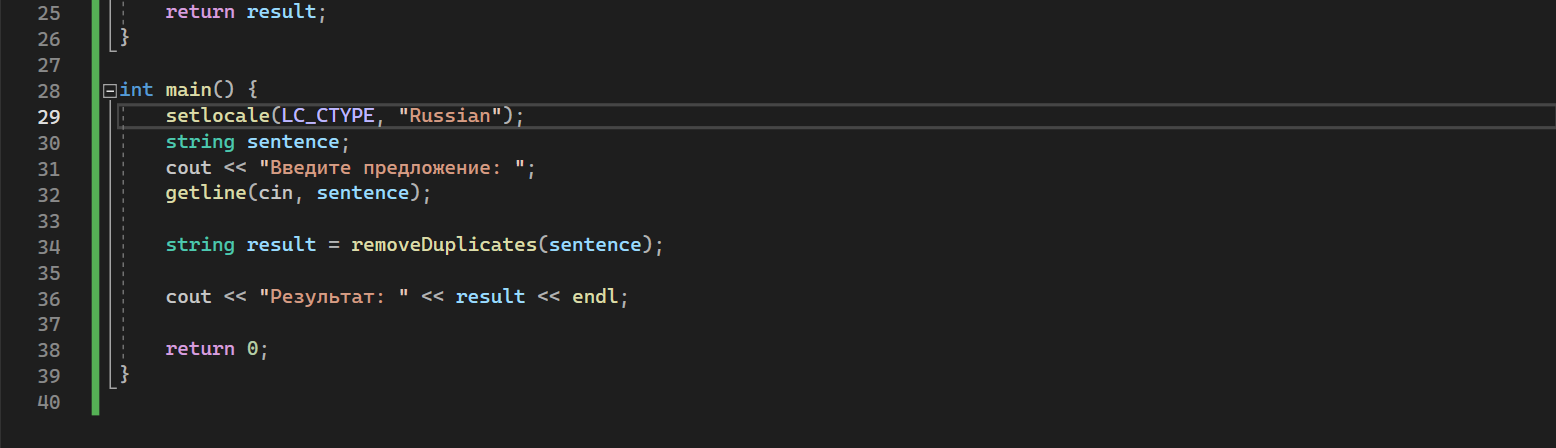
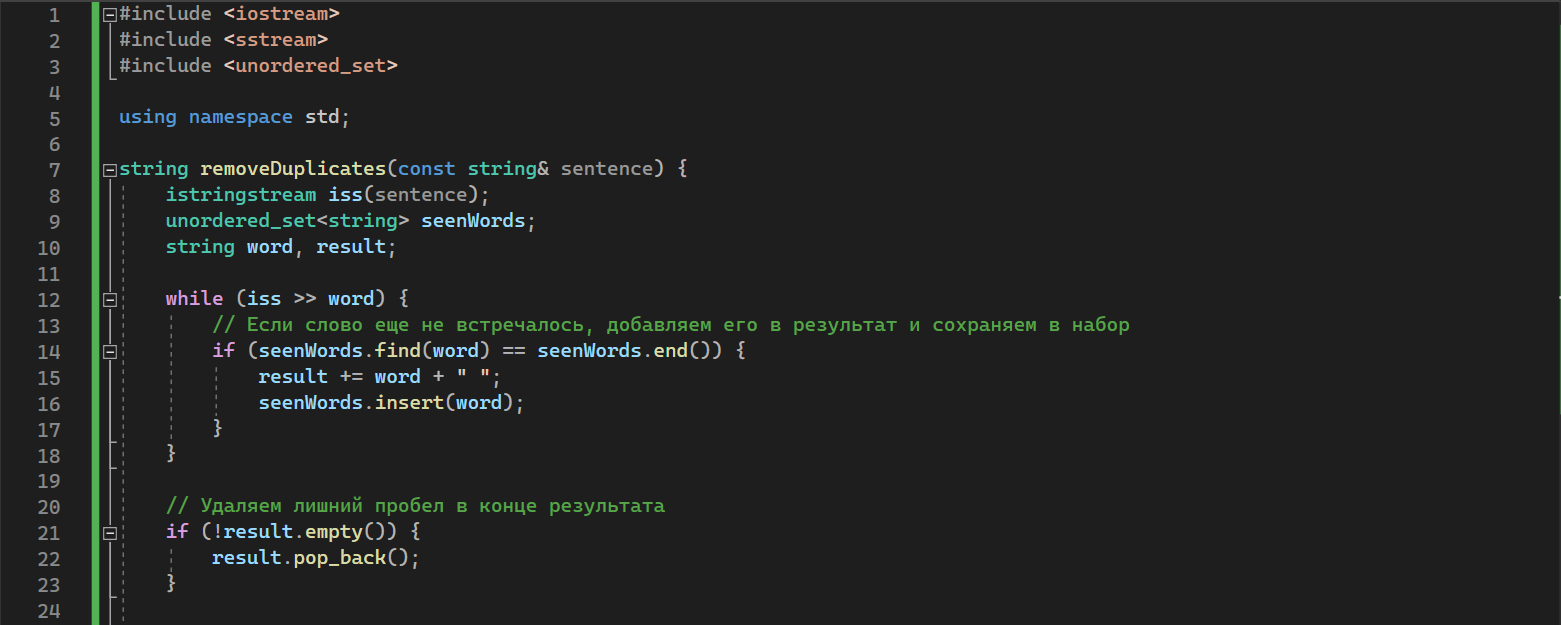
}

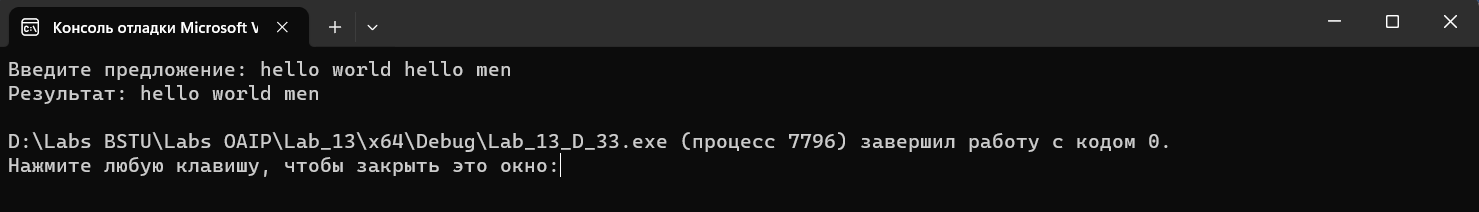
****

**Дополнительные задания:**

1. **Задание 3.**

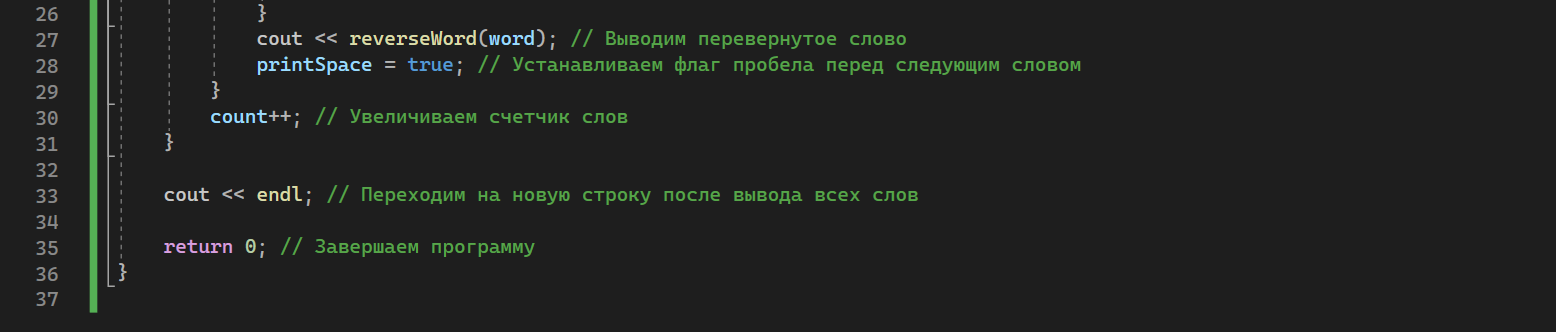
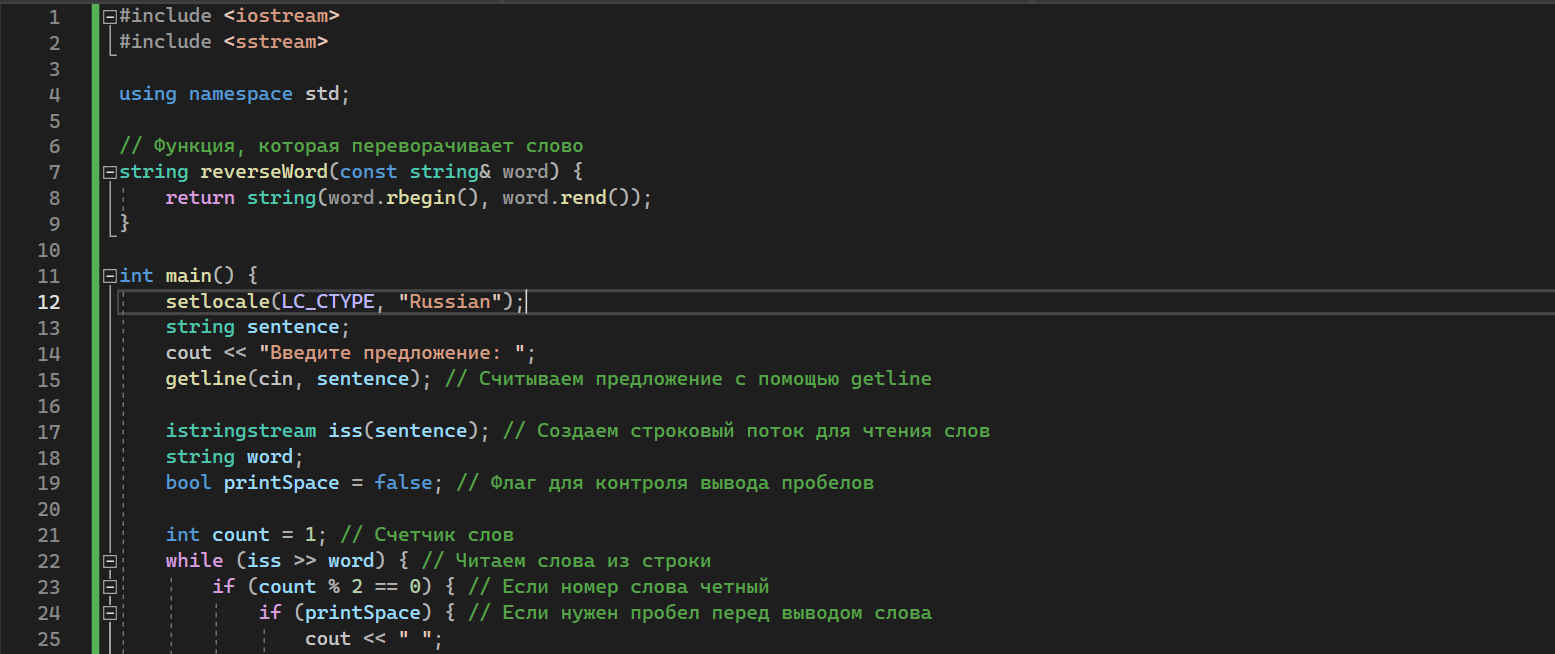
Из заданного предложения удалить те слова, которые уже встречались в предложении раньше.

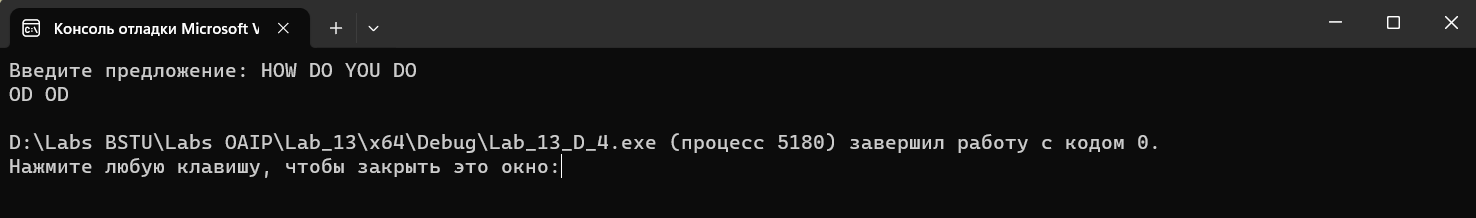




1. **Задание 4**

Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами. Пример: HOW DO YOU DO преобразовать в OD OD.





1. **Задание 5**

Даны два предложения. Найти самое короткое из слов первого предложения, которого нет во втором предложении.

