

Государственный университет Республики Молдова  
Факультет Математики и Информатики  
Департамент Информатики

## **Лабораторная работа №3**

по курсу “Основы программирования”

тема: Работа со строками (векторами символов)

Выполнил:  
студент группы I2302  
Славов Константин

Проверил преподаватель:  
Латул Г.

Кишинев, 2023

## Лабораторная работа №3

### Задание:

Отредактировать заданное предложение, заменяя его инвертированным (т.е. "читаемым" в обратном порядке). От себя я добавил небольшие корректировки задания, которые оставляют реверсивный текст под оригинальным предложением на своих местах.

### Листинг программы:

```
#include <iostream>
#include <sstream>
#include <algorithm>
#include <vector>

using namespace std;

int main() {

    string text;
    cout << "Enter your text here: " << endl;
    getline(cin, text); //Использую функцию getline
    для того, чтобы считывать введенный текст с
    клавиатуры и сохранения
    //ее в переменной text.

    istringstream stringText(text); //Создаю объект
    istringstream с названием stringText. Объект
    stringText считывает
    //данные со строки, как если бы они были вводом
    с клавиатуры, но вместо этого они считываются из
    строки inputText.

    vector <string> words; //Создается вектор words,
    в котором будут храниться отдельные слова из
    введенного текста.
```

```

    // Разбить текст на слова
    string word;
    while (stringText >> word) { //Цикл будет
выполняться, пока stringText успешно считывает слово
и присваивает его переменной word.
        words.push_back(word); //Внутри цикла каждый
раз, пока слово успешно считывается из stringText,
оно добавляется в конец
        //вектора words при помощи метода push_back.
Это расширяет вектор words, добавляя новое слово в
его конец на каждой итерации цикла.
    }

    // Инвертировать каждое слово
    for (string& word : words) { //Данная строка
использует ссылку на word, что означает что word
будет ссылаться на каждый элемент words,
        //а не создавать копии.
        reverse(word.begin(), word.end());
//Используется функция reverse для того, чтобы текст
из начала переносился в конец.
        //Для этого используются методы .begin и
.end.
    }

    // Вывести инвертированный текст
    for (const string& word : words) { //Здесь
каждый элемент будет поочередно ссылаться на
переменную word,
        //и эта переменная будет использоваться для
представления текущего элемента в векторе words.
        cout << word << " ";
    }

    cout << endl;

    return 0;
}

```

### **Используемые библиотеки:**

**iostream** - это стандартная библиотека в C++, которая предоставляет возможности для ввода и вывода данных.

**vector** - представляет собой динамический массив, который может изменять свой размер во время выполнения программы. Она используется для создания и управления динамическими массивами элементов одного типа.

**sstream** - это часть стандартной библиотеки C++, которая предоставляет возможность работы с потоками в памяти, а именно с потоками, связанными с строками. Этот класс позволяет вам выполнять различные операции с данными, как если бы они были потоками ввода и вывода, но вместо физических файлов, они манипулируют строками в памяти.

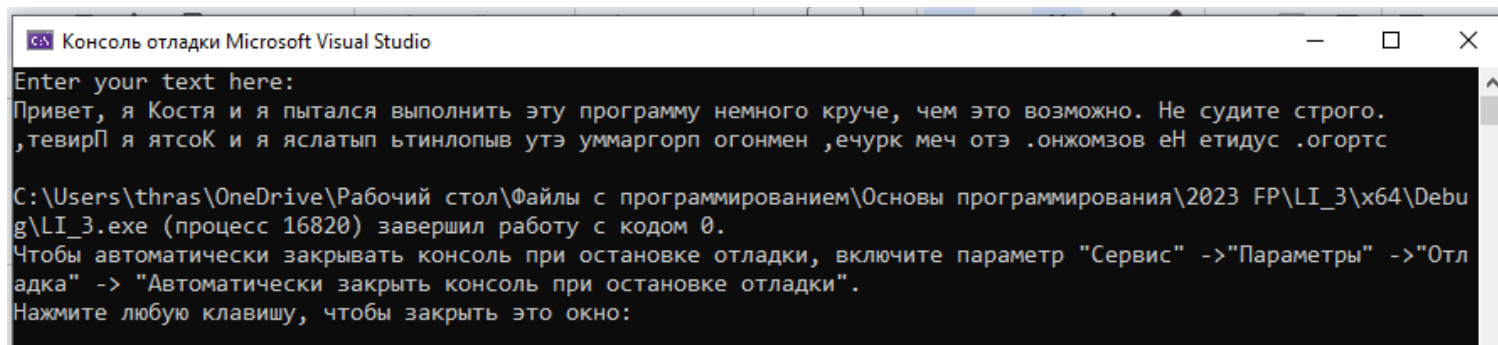
**algorithm** - это заголовочный файл в стандартной библиотеке C++, который содержит множество полезных алгоритмов для работы с контейнерами, итераторами и другими структурами данных. Эти алгоритмы предоставляют разнообразные операции, такие как сортировка, поиск, обход и преобразование элементов контейнеров.

### **Используемые функции:**

**getline** - это функция из стандартной библиотеки C++, предназначенная для считывания строк из потока ввода. Основным применением **getline** является считывание текста, введенного пользователем или данных из файла.

**reverse** - функция из стандартной библиотеки C++ используется для обращения порядка элементов в последовательности. Она меняет порядок элементов на обратный.

## Результат выполнения программы:



```
Консоль отладки Microsoft Visual Studio
Enter your text here:
Привет, я Костя и я пытался выполнить эту программу немного круче, чем это возможно. Не судите строго.
,тевирП я ятсоK и я яслатып ьтинлопыв утэ уммаргорп огонмен ,ечурк меч отэ .онжомзов eH етидус .огортс

C:\Users\thras\OneDrive\Рабочий стол\Файлы с программированием\Основы программирования\2023 FP\LI_3\x64\Debu
g\LI_3.exe (процесс 16820) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отл
адка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```