

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационных систем

ОТЧЕТ
по практической работе №4
по дисциплине «Программирование»
Тема: Текстовые строки как массивы символов

Студент гр. 1323

Князев И. А.

Преподаватель

Глущенко А. Г.

Санкт-Петербург

2022

Цель работы.

Изучение представления текстовых строк в программах. Работа со строками как с массивами символов. Операции над строками. Знакомство с классом **std::string**.

Основные теоретические положения.

Текстовые строки представляются с помощью одномерных массивов символов. В языке C++ текстовая строка представляет собой набор символов, обязательно заканчивающийся нулевым символом ('\0'). Поэтому, если вы хотите создать текстовый массив для хранения 10 (N) символов, нужно выделить память под 11 (N+1) символов.

Объявленный таким образом массив может использоваться для хранения текстовых строк, содержащих не более 10 символов. Нулевой символ позволяет определить границу между содержащимся в строке текстом и неиспользованной частью строки.

Класс **string** предназначен для работы со строками типа **char**, которые представляют собой строчку с завершающим нулем (символ '\0'). Класс **string** был введен как альтернативный вариант для работы со строками типа **char**.

Чтобы использовать возможности класса **string**, нужно подключить библиотеку **<string>** и пространство имен **std**. Объявление же переменной типа **string** осуществляется схоже с обычной переменной:

Постановка задачи.

Необходимо написать программу, которая:

1) С клавиатуры или с файла (пользователь сам может выбрать способ ввода) вводится последовательность, содержащая от 1 до 50 слов, в каждом из которых от 1 до 10 строчных латинских букв и цифр. Между соседними словами произвольное количество пробелов. За последним символом стоит точка.

2) Необходимо отредактировать входной текст:

- удалить лишние пробелы;
- удалить лишние знаки препинания (под «лишними» подразумевается несколько подряд идущих знаков (обратите внимание, что «...» - корректное использование знака) в тексте);
- исправить регистр букв, если это требуется (пример некорректного использования регистра букв: пРиМЕр);

3) Выполнить задание по варианту: вывести на экран слова последовательности в обратном порядке.

4) Выполнить задание по варианту: вывести на экран все слова последовательности в две или три колонки (в зависимости от количества слов) с выравниванием слов по правой границе колонки.

5) Необходимо найти все подстроки, которую введёт пользователь в имеющейся строке. Реализуйте два алгоритма: первый алгоритма – Линейный поиск, а второй алгоритм согласно вашему номеру в списке. Четные номера должны реализовать алгоритм КНМ, а нечетные – Бойера-Мура.

Выполнение работы.

Исходный код программы можно найти в репозитории:

<https://github.com/puppymail/lab>

Результаты работы программы.

```
PS C:\Users\acc13\Documents\cpp\lab\04> .\lab04.exe
1. Enter between 1 and 50 words between 1 and 10 symbols each separated by space and terminated by a dot
Please, enter text:
SOME      tExt.....  EXAmple L0000000000000000ONGWORD,,,, text.

2. Edit input text
Before: 'SOME      tExt.....  EXAmple L0000000000000000ONGWORD,,,, text.'
After: 'Some text... Example, text.'

3. Reverse words
After reverse: 'emoS txet... elpmaxE, txet.'

4. Print words in columns with right-alignment
      Some      text
Example      text

Press any key to continue . . .
```

Рисунок 1.1. Результат программы (ч.1).

Вывод.

Во время выполнения данной работы были изучены строки, массивы символов и принципы работы с ними. Также был получен навык работы со строками.