**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем (ИС)**

отчет

**по лабораторной работе №4**

**по дисциплине «Программирование»**

Тема: **Текстовые строки как массивы символов**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1323 |  | Лбова А. С. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2021

**Цель работы.**

Изучение способов обработки текстовых строк; изучение алгоритмов поиска подстроки в строке.**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’). 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. При определении строковых переменных их можно инициализировать

конкретными значениями с помощью строковых литералов.

Непосредственное чтение текстовых строк из потока вывода осуществляется до первого знака пробела. Для того чтобы прочесть всю строку полностью, можно воспользоваться одной из функций gets или gets\_s.

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.

Создание нового типа string было обусловлено недостатками работы со

строками символов, который показывал тип char. В сравнении с этим типом

string имеет ряд основных преимуществ:

1. возможность использования для обработки строк стандартные операторы С++ (=, +, <, ==, >, +=, ! =, <=, >=,[]). Использование типа char требует написания чрезмерного программного кода;
2. обеспечение лучшей надежности программного кода;
3. обеспечение строки как самостоятельного типа данных.

Но весь функционал string накладывает и свой негативный отпечаток. Основным недостатком string в сравнении с типом char является замедленная скорость обработки данных.

Самый простой способ подстроки в строке – Линейный поиск – циклическое сравнение всех символов строки с подстрокой. Действительно, этот способ первым приходит в голову, но очевидно, что он будет самым долгим.

Одним из самых популярных алгоритмов, который работает – быстрее,

чем приведенный ранее алгоритм, является алгоритм Кнута–Морриса–Пратта. Идея заключается в том, что не нужно проходить и сравнивать абсолютно все символы строки, если известны символы, которые есть и в

строке, и в подстроке.

Алгоритм Бойера–Мура в отличие от КМП полностью независим и не требует заранее проходить по строке. Этот алгоритм считается наиболее быстрым среди алгоритмов общего назначения, предназначенных для поиска подстроки в строке.

Преимущество этого алгоритма в том, что ценой некоторого количества

предварительных вычислений над подстрокой подстрока сравнивается с исходным текстом не во всех позициях (пропускаются позиции, которые точно не дадут положительный результат). Поиск подстроки ускоряется благодаря созданию таблиц сдвигов.

Правила построения таблицы сдвигов:

1. Определить значение элемента таблицы, которое равно удаленности соответствующего символа от конца шаблона (подстроки).
2. Если символ встречается более одного раза, то применятся значение, соответствующее символу, наиболее близкому к концу шаблона.
3. Если символ в конце шаблона встречается один раз, ему соответствует значение, равное длине образа, если более одного раза – значение, соответствующее символу, наиболее близкому к концу образа.
4. Для символов, отсутствующих в образе, применяется значение, равное длине шаблона.

**Ход работы**

1. Программа получает последовательность с помощью ввода с клавиатуры.
2. Идет редактирование текста, убирая лишние пробелы и знаки препинания (кроме «…»), и исправляя регистр, если это требуется.
3. Выводит последовательность, не содержащую цифры.
4. Выводит на экран ту же последовательность, заменяя во всех словах первую букву соответствующей прописной буквой.
5. Осуществляет линейный поиск подстроки в строке.

**Результат выполнения программы**

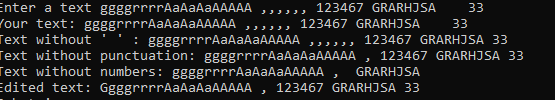


Рисунок 1



Рисунок 2

**Выводы.**

Были изучены способы обработки текстовых строк. Разобран алгоритм линейного поиска подстроки в строке.