Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский

Томский политехнический Университет»



Инженерная школа ядерных технологий

Направление 01.04.02 «Прикладная математика и информатика»

**ОТЧЕТ**

по лабораторной работе

Нормальные алгоритмы Маркова

по дисциплине:

**История и методология прикладной математики и информатики**

|  |  |  |  |
| --- | --- | --- | --- |
| **Исполнитель:** |  | Е. В. Петрович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| студент группы 0ВМ92 |  | Дата сдачи: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |  |
|  |  |  |  |
| **Руководитель:** |  | Ю. Б. Буркатовская | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| доцент, |  | Дата проверки: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| кандидат физико-математических наук |  |  |  |
|  |  |  |  |

Томск – 2020

Оглавление

[Задание 1 2](#_Toc47992856)

[Задание 2 2](#_Toc47992857)

[**Цель работы** 3](#_Toc47992858)

[**Основная часть** 3](#_Toc47992859)

[Задание 1 4](#_Toc47992860)

[Задание 2 7](#_Toc47992861)

[**Приложение 1** 9](#_Toc47992862)

[Ссылка на проект 9](#_Toc47992863)

**Задание**

Задание 1

Удвоить строку (например, 00102 превратить в 0010200102).

Задание 2

Проверить корректность записи арифметического выражения, например:

1. (x\*(x+x)+x)\*(x+x)+x - корректно
2. (x\*(x++x)+x)\*(x+x+x - некорректно

**Цель работы**

Изучить работу нормальных алгоритмов Маркова.

**Основная часть**

Нормальный алгоритм Маркова – способ формального определения понятия алгоритма, предложенный математиком Андреем Андреевичем Марковым (младшим) в 1947 году.

Нормальный алгоритм Маркова представляет собой способ преобразования слов – произвольных последовательностей символов, определенных в некотором алфавите A. Преобразование определяется упорядоченной последовательностью правил элементарных подстановок одной подстроки другой подстрокой. Обозначаются правила следующим образом:

α → β,

где α и β – подстроки, которые могут быть и пустыми.

Правила могут быть простыми и заключительными. Сразу после выполнения простого правила алгоритм продолжает работу, а после выполнения заключительного правила подстановки алгоритм останавливается.

Работа алгоритма происходит следующим образом:

1. Слово просматривается для нахождения в нем подстроки α. Если в слове присутствуют несколько вхождений подстроки α, то берется самое первое вхождение. Если подстрока α текущего правила подстановки не найдена, алгоритм переходит к следующему правилу подстановки.
2. Подстрока α заменяется подстрокой β.
3. После выполнения подстановки алгоритм снова переходит к первому правилу из последовательности правил подстановки.
4. Алгоритм продолжается пока не будет выполнено заключительное правило или пока не будет найдено ни одной подстроки α из всех правил подстановки, входящих в последовательность.

Задание 1

Удвоить строку (например, 00102 превратить в 0010200102).

Для реализации алгоритма вводим алфавит, состоящий из цифр от 0 до 9 (обозначение в блок-схеме алгоритма d), букв от ‘a’ до ‘j’ (обозначение в блок-схеме алгоритма c) и знаков ‘<’ и ‘>’. Каждой цифре сопоставим соответствующую ей по порядку букву латинского алфавита (*Таблица 1*).

|  |  |  |
| --- | --- | --- |
|  |  | *Таблица 1* |
|  | Соответствие букв цифрам |  |

|  |  |
| --- | --- |
| Цифра d | Буква c |
| 1 | a |
| 2 | b |
| 3 | c |
| 4 | d |
| 5 | e |
| 6 | f |
| 7 | g |
| 8 | h |
| 9 | i |
| 0 | j |

Описание алгоритма удвоения строки:

* Каждая цифра строки заменяется на пару символов , где – соответствующая буква (см. Таблица 1);
* Все буквы смещаются в конец строки путем замены подстрок на подстроки ;
* Все буквы замещаются на соответствующие им цифры (см. Таблица 1).

Реализация алгоритма удвоения представлена на Рис.1. Блок-схема алгоритма представлена на Рис.2. Ссылка на файл представлена в Приложении 1.

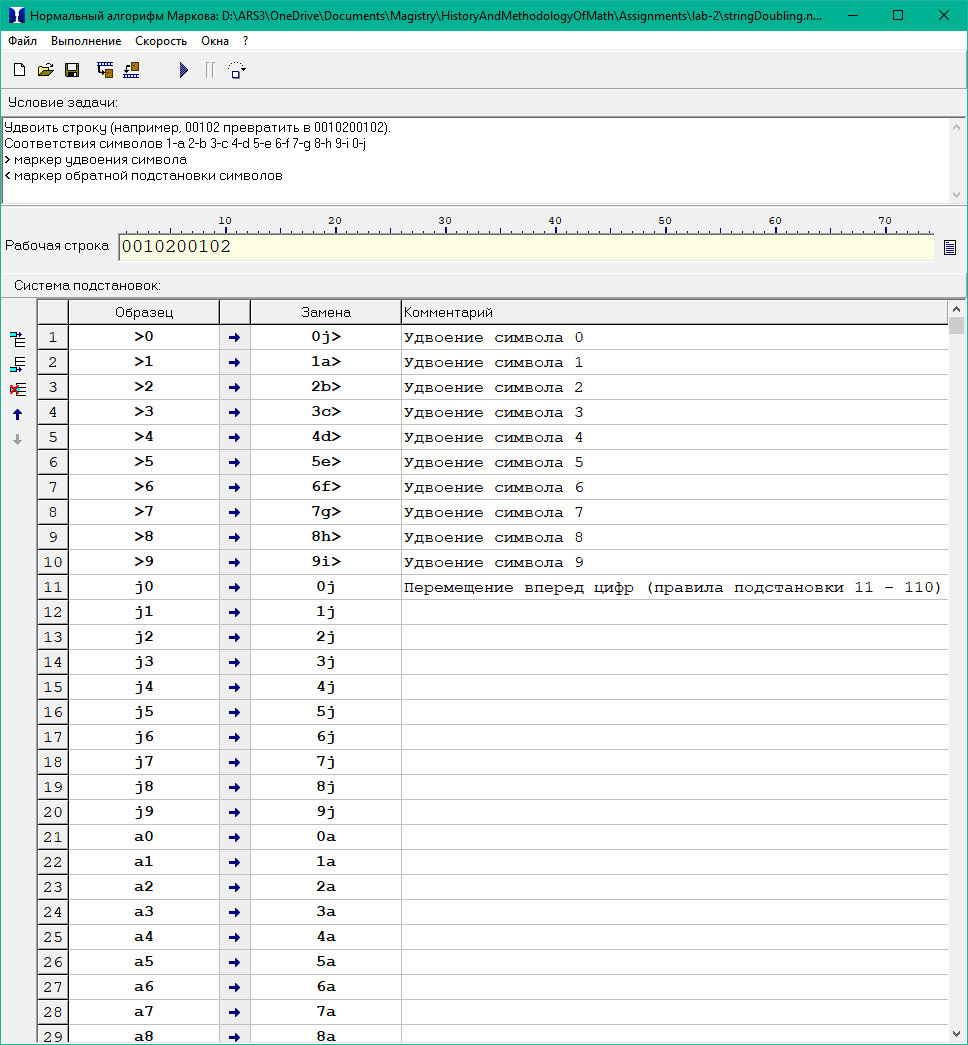


Рис.1 Реализация алгоритма удвоения строки.



Рис. 2 Блок-схема алгоритма удвоения строки

Задание 2

Проверить корректность записи арифметического выражения, например:

1. (x\*(x+x)+x)\*(x+x)+x - корректно
2. (x\*(x++x)+x)\*(x+x+x - некорректно

Для реализации алгоритма вводим алфавит {x+\*-/()}. Алгоритм проверки представляет собой последовательные подстановки корректных подстрок символом ‘x’. Если выражение корректно, то результирующим словом должно состоять из единственного символа ‘x’.

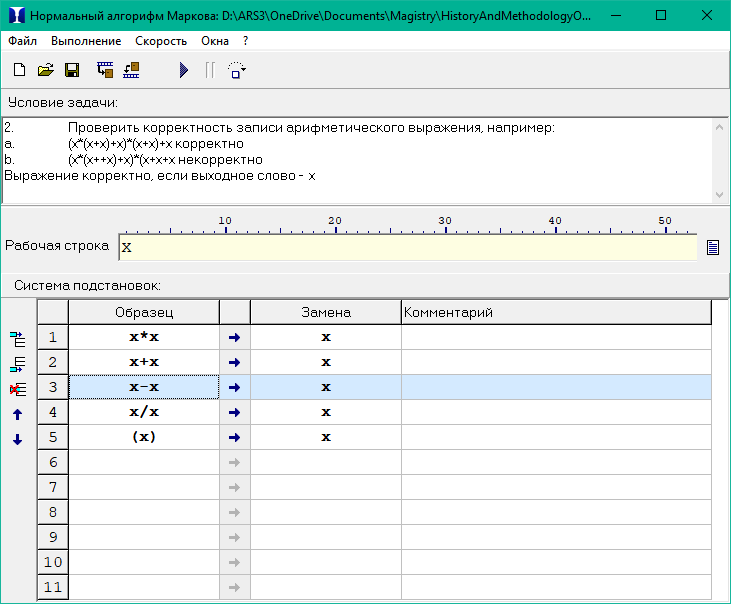


Рис.3 Реализация алгоритма проверки корректности выражения.

**Заключение**

В работе разработаны и реализованы алгоритмы удвоения строки и проверки выражения на корректность. Представлены скриншоты работы нормальных алгоритмов Маркова, блок-схемы алгоритмов, файлы реализации нормальных алгоритмов Маркова.

**Приложение 1**

Ссылка на проект

<https://github.com/petrovicheugene/MethodologyOfAppliedMathematics.git>