**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Информатика»**

Тема: **Машина Тьюринга**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3342 |  | Галеев А.Д. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Изучить понятие машины Тьюринга и узнать о принципе ее работы.

## Задание

На вход программе подается строка неизвестной длины. Каждый элемент является значением в ячейке памяти ленты Машины Тьюринга. На ленте находится последовательность латинских букв из алфавита a, b, c. Напишите программу, которая заменяет в исходной строке символ, предшествующий первому встретившемуся символу c на символ, следующий за первым встретившимся символом a. Если первый встретившийся символ a в конце строки, то используйте его в качестве заменяющего. Указатель на текущее состояние Машины Тьюринга изначально находится слева от строки с символами (но не на первом ее символе). По обе стороны от строки находятся пробелы.

Соглашения:

1. Направление движения автомата может быть одно из R (направо), L (налево), N (неподвижно).

2. Гарантируется, что длинна строки не менее 5 символов и не более 15.

3. В середине строки не могут встретиться пробелы.

4. При удалении или вставке символов направление сдвигов подстрок не принципиально (т. е. результат работы алгоритма может быть сдвинут по ленте в любую ее сторону на любое число символов).

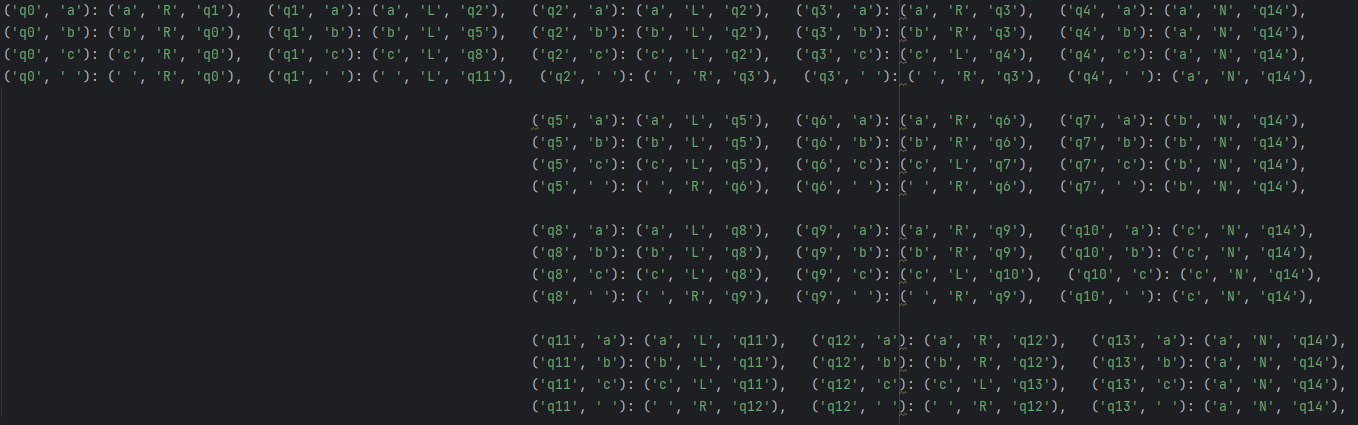
5. Курсор по окончании работы алгоритма может находиться на любом символе.

## Основные теоретические положения

Для решения задач в программе использовались только функции из стандартной библиотеки языка Python.

## Выполнение работы

Программа состоит из функции turing\_machine, которая принимает входные данные и возвращает результат, после чего он выводится в консоль.

Функция turing\_machine работает по принципу машины Тьюринга, сначала полученная строка записывается в список tape, первым элементом которого является пробел, т.к. по условию сначала коронка находится левее от начала ленты. Так же обозначается начальное положение коронки и начальное состояние q0. Всего получилось 14 состояний:

Состояние q0 отвечает за поиск первого символа ’a‘, после его нахождения, коронка переходит в состояние q1, в зависимости от символа который идет после ‘a’ коронка пойдет по одной из четырех веток, все ветки работают по одному принципу поэтому расскажу о принципе работы ветки q2-q4. Коронка перейдет на ветку q2 только в том случае если после первого символа ‘a’ в строке идет символ ’a’, в состоянии q2 коронка возвращается в начальное положение, после этого она переходит в состояние q3 в котором она ищет первый символ ’c’, после его нахождения она уходит на символ влево, и переходит в состояние q4 где заменяет этот символ на ’a’, после чего завершает путь.

**Тестирование**

Результаты тестирования представлены в табл. 1  
Таблица 1 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| № проверки | Входные данные | Выходные данные |
| 1. | abcabacbabcab | abcabacbabcab |
| 2. | cbabcab | bcbabcab |
| 3. | ccccaab | accccaab |

## Выводы

Было изучено понятие машины Тьюринга.

Разработана программа выполняющая замену символов с помощью алгоритма машины Тьюринга.

# Приложение А Исходный код программы

Название файла: main\_lb3

def turing\_machine(input\_tape):  
 input\_tape = ' ' + input\_tape  
 tape = list(input\_tape)  
 state = 'q0'  
 head\_position = 0  
 states\_history = [state]  
 transitions = {  
 ('q0', 'a'): ('a', 'R', 'q1'), ('q1', 'a'): ('a', 'L', 'q2'), ('q2', 'a'): ('a', 'L', 'q2'), ('q3', 'a'): ('a', 'R', 'q3'), ('q4', 'a'): ('a', 'N', 'q14'),  
 ('q0', 'b'): ('b', 'R', 'q0'), ('q1', 'b'): ('b', 'L', 'q5'), ('q2', 'b'): ('b', 'L', 'q2'), ('q3', 'b'): ('b', 'R', 'q3'), ('q4', 'b'): ('a', 'N', 'q14'),  
 ('q0', 'c'): ('c', 'R', 'q0'), ('q1', 'c'): ('c', 'L', 'q8'), ('q2', 'c'): ('c', 'L', 'q2'), ('q3', 'c'): ('c', 'L', 'q4'), ('q4', 'c'): ('a', 'N', 'q14'),  
 ('q0', ' '): (' ', 'R', 'q0'), ('q1', ' '): (' ', 'L', 'q11'), ('q2', ' '): (' ', 'R', 'q3'), ('q3', ' '): (' ', 'R', 'q3'), ('q4', ' '): ('a', 'N', 'q14'),  
  
 ('q5', 'a'): ('a', 'L', 'q5'), ('q6', 'a'): ('a', 'R', 'q6'), ('q7', 'a'): ('b', 'N', 'q14'),  
 ('q5', 'b'): ('b', 'L', 'q5'), ('q6', 'b'): ('b', 'R', 'q6'), ('q7', 'b'): ('b', 'N', 'q14'),  
 ('q5', 'c'): ('c', 'L', 'q5'), ('q6', 'c'): ('c', 'L', 'q7'), ('q7', 'c'): ('b', 'N', 'q14'),  
 ('q5', ' '): (' ', 'R', 'q6'), ('q6', ' '): (' ', 'R', 'q6'), ('q7', ' '): ('b', 'N', 'q14'),  
  
 ('q8', 'a'): ('a', 'L', 'q8'), ('q9', 'a'): ('a', 'R', 'q9'), ('q10', 'a'): ('c', 'N', 'q14'),  
 ('q8', 'b'): ('b', 'L', 'q8'), ('q9', 'b'): ('b', 'R', 'q9'), ('q10', 'b'): ('c', 'N', 'q14'),  
 ('q8', 'c'): ('c', 'L', 'q8'), ('q9', 'c'): ('c', 'L', 'q10'), ('q10', 'c'): ('c', 'N', 'q14'),  
 ('q8', ' '): (' ', 'R', 'q9'), ('q9', ' '): (' ', 'R', 'q9'), ('q10', ' '): ('c', 'N', 'q14'),  
  
 ('q11', 'a'): ('a', 'L', 'q11'), ('q12', 'a'): ('a', 'R', 'q12'), ('q13', 'a'): ('a', 'N', 'q14'),  
 ('q11', 'b'): ('b', 'L', 'q11'), ('q12', 'b'): ('b', 'R', 'q12'), ('q13', 'b'): ('a', 'N', 'q14'),  
 ('q11', 'c'): ('c', 'L', 'q11'), ('q12', 'c'): ('c', 'L', 'q13'), ('q13', 'c'): ('a', 'N', 'q14'),  
 ('q11', ' '): (' ', 'R', 'q12'), ('q12', ' '): (' ', 'R', 'q12'), ('q13', ' '): ('a', 'N', 'q14'),  
 }  
 while state != 'q14':  
 symbol = tape[head\_position]  
 new\_symbol, move, new\_state = transitions.get((state, symbol), (' ', 'N', 'q14'))  
 tape[head\_position] = new\_symbol  
 if move == 'R' and head\_position < len(tape) - 1:  
 head\_position += 1  
 elif move == 'L' and head\_position > 0:  
 head\_position -= 1  
 state = new\_state  
 states\_history.append(state)  
 result\_tape = ''.join(tape)  
  
 return result\_tape  
  
input\_tape = input()  
result\_tape = turing\_machine(input\_tape)  
print(result\_tape)