**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. 3343 |  | Атоян М. А. |
| Преподаватель |  | Иванов Д. В. |

Санкт-Петербург

2023

**Цель работы**

Изучить принцип работы машины Тьюринга и научиться его применять для решения задач.

**Задание**

Вариант 3.

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, идущий после последних двух встретившихся символов 'a', на предшествующий им символ(гарантируется, что это не пробел). Наличие в строке двух подряд идущих символов 'a' гарантируется.

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.

Алфавит:

* a
* b
* c
* " " (пробел)

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.

Ваша программа должна вывести полученную ленту после завершения работы.

В отчет включите таблицу состояний. Отдельно кратко опишите каждое состояние, например:

q1 - начальное состояние, которое необходимо, чтобы найти первый встретившийся символ ‘c’.

## Выполнение работы

Таблица состояний машины Тьюринга представлена в табл. 1.

Таблица 1 – Состояния машины Тьюринга

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Состояние | | a | b | c | " " |
| q0 | | 'a', N, 'q1' | 'b', N, 'q1' | 'c', N, 'q1' | ' ', R, 'q0' |
| q1 | | 'a', R, 'q1' | 'b', R, 'q1' | 'c', R, 'q1' | ' ', L, 'q2' |
| q2 | | 'a', L, 'q3' | 'b', L, 'q2' | "c", "L", "q9" |  |
| q3 | 'a', L, 'q4' | | 'b', L, 'q2' | 'c', L, 'q2' |  |
| q4 | | 'a', R, 'q5' | 'b', R, 'q7' | 'c', R, 'q9' | ' ', R, 'q11' |
| q5 | | 'a', R, 'q5' | 'b', N, 'q6' | 'c', N, 'q6' | 'a', N, 'q6' |
| q6 | | 'a', N, 'qF' | 'a', N, 'qF' | 'a', N, 'qF' | 'a', N, 'qF' |
| q7 | | 'a', R, 'q7' | 'b', N, 'q8' | 'c', N, 'q8' | 'a', N, 'q8' |
| q8 | | 'b', N, 'qF' | 'b', N, 'qF' | 'b', N, 'qF' | 'b', N, 'qF' |
| q9 | | 'a', R, 'q9' | 'b', N, 'q10' | 'c', N, 'q10' | 'a', N, 'q10' |
| q10 | | 'c', N, 'qF' | 'c', N, 'qF' | 'c', N, 'qF' | 'c', N, 'qF' |
| q11 | | 'a', R, 'q11' | 'b', N, 'q12' | 'c', N, 'q12' | ' ', R, 'q12' |
| q12 | | ' ', N, 'qF' | ' ', N, 'qF' | ' ', N, 'qF' | ' ', N, 'qF' |

Описание состояний:

q1 — начальное положение, поиск первого символа а

q2 — найден символ следующий за первым встретившимся символом а

q3 — предшествующим был символ b, возвращение в начало ленты

q4 — поиск символа с

q5 — заменяется символ, предшествующий символу с, на b

q6 — предшествующим был символ а, возвращение в начало ленты

q7 — поиск символа с

q8 — заменяется символ, предшествующий символу с, на а

q9 — предшествующим был символ с, возвращение в начало ленты

q10 — поиск символа с

q11 — заменяется символ, предшествующий символу с, на с

q12 — предшествующим был символ " ", возвращение в начало ленты

q13 — поиск символа с

q14 — заменяется символ, предшествующий символу с, на а

Разработанный программный код см. в приложении А.

## Тестирование

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | |  |  | | --- | --- | | abcaabc |  | | abcaacc | Выходные данные соответствуют ожиданиям. |
|  | aabbaa | aabbaab | Выходные данные соответствуют ожиданиям. |
|  | aabb | aa b | Выходные данные соответствуют ожиданиям. |

## Выводы

В ходе выполнения лабораторной работы были изучены и освоены необходимые навыки для создания машины Тьюринга.

# Приложение А Исходный код программы

Название файла: main.py

R, L, N = 1, -1, 0

table = {

'q0': {'a': ['a', N, 'q1'], 'b': ['b', N, 'q1'], 'c': ['c', N, 'q1'], ' ': [' ', R, 'q0']},

'q1': {'a': ['a', R, 'q1'], 'b': ['b', R, 'q1'], 'c': ['c', R, 'q1'], ' ': [' ', L, 'q2']},

'q2': {'a': ['a', L, 'q3'], 'b': ['b', L, 'q2'], 'c': ['c', L, 'q2']},

'q3': {'a': ['a', L, 'q4'], 'b': ['b', L, 'q2'], 'c': ['c', L, 'q2']},

'q4': {'a': ['a', R, 'q5'], 'b': ['b', R, 'q7'], 'c': ['c', R, 'q9'], ' ': [' ', R, 'q11']},

'q5': {'a': ['a', R, 'q5'], 'b': ['b', N, 'q6'], 'c': ['c', N, 'q6'], ' ': ['a', N, 'q6']},

'q6': {'a': ['a', N, 'qF'], 'b': ['a', N, 'qF'], 'c': ['a', N, 'qF'], ' ': ['a', N, 'qF']},

'q7': {'a': ['a', R, 'q7'], 'b': ['b', N, 'q8'], 'c': ['c', N, 'q8'], ' ': ['a', N, 'q8']},

'q8': {'a': ['b', N, 'qF'], 'b': ['b', N, 'qF'], 'c': ['b', N, 'qF'], ' ': ['b', N, 'qF']},

'q9': {'a': ['a', R, 'q9'], 'b': ['b', N, 'q10'], 'c': ['c', N, 'q10'], ' ': ['a', N, 'q10']},

'q10': {'a': ['c', N, 'qF'], 'b': ['c', N, 'qF'], 'c': ['c', N, 'qF'], ' ': ['c', N, 'qF']},

'q11': {'a': ['a', R, 'q11'], 'b': ['b', N, 'q12'], 'c': ['c', N, 'q12'], ' ': [' ', R, 'q12']},

'q12': {'a': [' ', N, 'qF'], 'b': [' ', N, 'qF'], 'c': [' ', N, 'qF'], ' ': [' ', N, 'qF']},

}

tape = ' ' + input() + ' '

state = 'q0'

pos = 0

while state!='qF':

commands = table[state]

commands = commands[tape[pos]]

tape = list(tape)

tape[pos] = commands[0]

tape = ''.join(tape)

pos += commands[1]

state = commands[2]

print(tape)