**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 |  | Отмахов Д. В. |
| Преподаватель |  | Иванов Д. В. |

Санкт-Петербург

2023

**Цель работы**

Изучить принцип работы машины Тьюринга, реализовать ее на языке Python.

**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, 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.

Для примера выше лента будет выглядеть так:

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 представлена в табл. 1.

Таблица 1 – Таблица состояний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 'a' | 'b' | 'c' | ' ' |
| q1 | 'a', R, 'q2' | 'b', R, 'q1' | 'c', R, 'q1' | ' ', R, 'q1' |
| q2 | 'a', L, 'q3' | 'b', L, 'q4' | 'c', L, 'q5' | ' ', L, 'q3' |
| q3 | 'a', L, 'q3' | 'b', L, 'q3' | 'c', L, 'q3' | ' ', R, 'q9' |
| q4 | 'a', L, 'q4' | 'b', L, 'q4' | 'c', L, 'q4' | ' ', R, 'q10' |
| q5 | 'a', L, 'q5' | 'b', L, 'q5' | 'c', L, 'q5' | ' ', R, 'q11' |
| q6 | 'a', N, 'qT' | 'a', N, 'qT' | 'a', N, 'qT' | 'a', N, 'qT' |
| q7 | 'b', N, 'qT' | 'b', N, 'qT' | 'b', N, 'qT' | 'b', N, 'qT' |
| q8 | 'c', N, 'qT' | 'c', N, 'qT' | 'c', N, 'qT' | 'c', N, 'qT' |
| q9 | 'a', R, 'q9' | 'b', R, 'q9' | 'c', L, 'q6' |  |
| q10 | 'a', R, 'q10' | 'b', R, 'q10' | 'c', L, 'q7' |  |
| q11 | 'a', R, 'q11' | 'b', R, 'q11' | 'c', L, 'q8' |  |

Описание состояний:

* q1 – начальное состояние, находит первый встретившийся символ ‘a’;
* q2 – состояние, определяет символ, следующий за первым встретившимся символом ‘a’;
* q3 – состояние для символа ‘a’, возвращает в начало строки;
* q4 – состояние для символа ‘b’, возвращает в начало строки;
* q5 – состояние для символа ‘c’, возвращает в начало строки;
* q6 – заменяет символ, предшествующий первому встретившемуся символу ‘c' на символ ‘a’;
* q7 – заменяет символ, предшествующий первому встретившемуся символу ‘c' на символ ‘b’;
* q8 – заменяет символ, предшествующий первому встретившемуся символу ‘c' на символ ‘c’;
* q9 – находит первый встретившийся символ ‘с’ и впоследствии заменяет предшествующий ему символ на символ ‘a’;
* q10 – находит первый встретившийся символ ‘с’ и впоследствии заменяет предшествующий ему символ на символ ‘b’;
* q11 – находит первый встретившийся символ ‘с’ и впоследствии заменяет предшествующий ему символ на символ ‘c’;
* qT – конечное состояние.

Описание переменных:

* tape – введенная строка;
* cur\_state – текущее состояние;
* index – индекс текущей ячейки.

Разработанный программный код см. в приложении А.

## Тестирование

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Выходные данные | Комментарий |
| 1. | abcabc | abcabc |  |
| 2. | acbacbcc | ccbacbcc |  |
| 3. | acbbcaacaa | ccbbcaacaa |  |

## Выводы

Был изучен принцип работы машины Тьюринга. Также была реализована программа на языке Python, выполняющая ее работу.

# Приложение А Исходный код программы

Название файла: main.py

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

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

'q2':{'a':['a', L, 'q3'], 'b':['b', L, 'q4'], 'c':['c', L, 'q5'], ' ':[' ', L, 'q3']},

'q3':{'a':['a', L, 'q3'], 'b':['b', L, 'q3'], 'c':['c', L, 'q3'], ' ':[' ', R, 'q9']},

'q4':{'a':['a', L, 'q4'], 'b':['b', L, 'q4'], 'c':['c', L, 'q4'], ' ':[' ', R, 'q10']},

'q5':{'a':['a', L, 'q5'], 'b':['b', L, 'q5'], 'c':['c', L, 'q5'], ' ':[' ', R, 'q11']},

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

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

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

'q9':{'a':['a', R, 'q9'], 'b':['b', R, 'q9'], 'c':['c', L, 'q6']},

'q10':{'a':['a', R, 'q10'], 'b':['b', R, 'q10'], 'c':['c', L, 'q7']},

'q11':{'a':['a', R, 'q11'], 'b':['b', R, 'q11'], 'c':['c', L, 'q8']}}

tape = list(input())

cur\_state = 'q1'

index = 0

while cur\_state != 'qT':

symbol = tape[index]

new\_symbol, move, cur\_state = table[cur\_state][symbol]

tape[index] = new\_symbol

index += move

print(''.join(tape))