**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. 3344 |  | Пачев Д.К. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

**Цель работы**

Написать программу на языке Python, которая имитирует работу машины Тьюринга.

**Задание**

### Вариант 2. 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.**

## **Выполнение работы**

Переменные L, R, N отвечают за то, куда нужно переместиться курсору на ленте. Создается переменная table, в которой хранится таблица состояний в виде словаря. Далее в переменную tape считывается входная строка. В переменных state, head и direction задаются соответственно начальное состояние q1, начальный индекс и направление движения курсора. Затем с помощью цикла while и таблицы состояний table происходила замена символов строки, идущих после двух “a”, на символ, предшествующий им.

Описание состояний:

q1 – состояние для нахождения начала строки

q2 – состояние для нахождения конца строки

q3 – состояние для нахождения первой с конца буквы “a”

q4 – состояние для нахождения повторного символа “a”

q5 – состояние для считывания символа, предшествующего “aa”

q6 – состояние для замены символа после aa на предшествующий им символ, если этот символ равен “a”

q7 – состояние для замены символа после aa на предшествующий им символ, если этот символ равен “b”

q8 – состояние для замены символа после aa на предшествующий им символ, если этот символ равен “c”

qn – конечное состояние

В конце программа выводит измененную ленту.

**Тестирование**

Результаты тестирования представлены в Таблице 1

Таблица 1 - Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
| 1. | abcaabc | abcaacc | - |
| 2. | aabbaa | aabbaab | - |

**Выводы**

В ходе выполнения лабораторной работы была разработана программа на языке Python, которая имитирует машину Тьюринга и выполняет поставленную задачу.

# ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

L = -1  
R = 1  
N = 0  
table = {  
 'q1': {' ': [' ', R, 'q1'], 'a': ['a', R, 'q2'], 'b': ['b', R, 'q2'], 'c': ['c', R, 'q2']},  
 'q2': {' ': [' ', L, 'q3'], 'a': ['a', R, 'q2'], 'b': ['b', R, 'q2'], 'c': ['c', R, 'q2']},  
 'q3': {'a': ['a', L,'q4'], 'b': ['b', L, 'q3'], 'c': ['c', L, 'q3']},  
 'q4': {'a': ['a', L, 'q5'], 'b': ['b', L, 'q3'], 'c': ['c', L, 'q3']},  
 'q5': {'a': ['a', R, 'q6'], 'b': ['b', R, 'q7'], 'c': ['c', R, 'q8']},  
 'q6': {' ': ['a', N, 'qn'], 'a': ['a', R, 'q6'], 'b': ['a', N, 'qn'], 'c': ['a', N, 'qn']},  
 'q7': {' ': ['b', N, 'qn'], 'a': ['a', R, 'q7'], 'b': ['b', N, 'qn'], 'c': ['b', N, 'qn']},  
 'q8': {' ': ['c', N, 'qn'], 'a': ['a', R, 'q8'], 'b': ['c', N, 'qn'], 'c': ['c', N, 'qn']}  
  
}  
tape = list(input())  
state = 'q1'  
head = 0  
direction = 0  
while state!='qn':  
 tape[head], direction, state = table[state][tape[head]]  
 head+=direction  
print(''.join(tape))