**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Информатика»**

Тема:Машина Тьюринга

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3342 |  | Шушко Л.Д. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Цель работы – изучение работы 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

Таблица состояний в коде написана в виде словаря. С помощью цикла while проверяется, не находится ли состояние в конечном виде, и до тех пор, пока не находится, выполняются инструкции, написанные в таблице состояний.

Таблица 1 — Таблица состояний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Состояние | «a» | «b» | «c» | « » |
| q1 | («a», 1, «q2») | («b», 1, «q2») | («c», 1, «q2») | (« », 1, «q1») |
| q2 | («a», 1, «q2») | («b», 1, «q2») | («c», 1, «q2») | (« », -1, «q3») |
| q3 | («a», -1, «q4») | («b», -1, «q3») | («c», -1, «q3») |  |
| q4 | («a», -1, «q5») | («b», -1, «q3») | («b», -1, «q3») |  |
| q5 | («a», 1, «q6») | («b», 1, «q7») | («c», 1, «q8») | (« », 1, «q9») |
| q6 | («a», 1, «q6») | («a», 0, «q10») | («a», 0, «q10») | («a», 0, «q10») |
| q7 | («a», 1, «q7») | («b», 0, «q10») | («b», 0, «q10») | («b», 0, «q10») |
| q8 | («a», 1, «q8») | («c», 0, «q10») | («c», 0, «q10») | («c», 0, «q10») |
| q9 | («a», 1, «q9») | (« », 0, «q10») | (« », 0, «q10») | (« », 0, «q10») |

q1 — начальное состояние

q2 — нахождение крайнего непустого символа

q3 — нахождение первого с конца символа «a»

q4 — проверка является ли следующая с конца буква символом «a»

q5 — переход на состояние, соответствующее символу, предшествующему двум «a»

q6 — перезапись следующего символа после двух «a» на «a»

q7 — перезапись следующего символа после двух «a» на «b»

q8 — перезапись следующего символа после двух «a» на «c»

q9 — перезапись следующего символа после двух «a» на « »

q10 — конечное состояние

Разработанный программный код см. в приложении А.

## Тестирование

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| № п/п | Входные данные | Выходные данные |
|  | abcaabc | abcaacc |
|  | aabbaa | aabbaab |

## Выводы

Изучена работа машины Тьюринга.

Разработана программа, имитирующая работу машины Тьюринга.

# Приложение А Исходный код программы

Название файла: Shushko\_Leya\_lb3.py

memory = list(input())

index = 0

state = 'q1'

table = {

'q1':{

'a':('a',1,'q2'),

'b':('b',1,'q2'),

'c':('c',1,'q2'),

' ':(' ',1,'q1'),

},

'q2':{

'a':('a',1,'q2'),

'b':('b',1,'q2'),

'c':('c',1,'q2'),

' ':(' ',-1,'q3'),

},

'q3':{

'a':('a',-1,'q4'),

'b':('b',-1,'q3'),

'c':('c',-1,'q3'),

},

'q4':{

'a':('a',-1,'q5'),

'b':('b',-1,'q3'),

'c':('c',-1,'q3'),

},

'q5':{

'a':('a',1,'q6'),

'b':('b',1,'q7'),

'c':('c',1,'q8'),

' ':(' ',1,'q9'),

},

'q6':{

'a':('a',1,'q6'),

'b':('a',0,'q10'),

'c':('a',0,'q10'),

' ':('a',0,'q10'),

},

'q7':{

'a':('a',1,'q7'),

'b':('b',0,'q10'),

'c':('b',0,'q10'),

' ':('b',0,'q10'),

},

'q8':{

'a':('a',1,'q8'),

'b':('c',0,'q10'),

'c':('c',0,'q10'),

' ':('c',0,'q10'),

},

'q9':{

'a':('a',1,'q9'),

'b':(' ',0,'q10'),

'c':(' ',0,'q10'),

' ':(' ',0,'q10'),

}

}

while state !='q10':

symbol = memory[index]

new\_symbol, delta, state = table[state][symbol]

memory[index]= new\_symbol

index+=delta

print(''.join(memory))