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 _____ Пухов А.Д.

Преподаватель _____ Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Реализация машины Тьюринга на Python.

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}.

			a	c	c	a	b	c	b	a	b	a	a	c	a	b			

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:

			c	c	c	a	b	c	b	a	b	a	a	c	a	b			

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 1 – таблица состояний

	‘a’	‘b’	‘c’	‘ ‘
q0	‘a’, R, ‘q1’	‘b’, R, ‘q0’	‘c’, R, ‘q0’	‘ ‘, R, ‘q0’
q1	‘a’, L, ‘qa’	‘b’, L, ‘qb’	‘c’, L, ‘qc’	‘ ‘, L, ‘qa_end’
qa	‘a’, L, ‘qa’	‘b’, L, ‘qa’	‘c’, L, ‘qa’	‘ ’, R, ‘qa1’
qa1	‘a’, R, ‘qa1’	‘b’, R, ‘qa1’	‘c’, L, ‘qac’	
qac	‘a’, N, ‘qT’	‘a’, N, ‘qT’		‘a’, N, ‘qT’
qb	‘a’, L, ‘qb’	‘b’, L, ‘qb’	‘c’, L, ‘qb’	‘ ’, R, ‘qb1’
qb1	‘a’, R, ‘qb1’	‘b’, R, ‘qb1’	‘c’, L, ‘qbc’	
qbc	‘b’, N, ‘qT’	‘b’, N, ‘qT’		‘b’, N, ‘qT’
qc	‘a’, L, ‘qc’	‘b’, L, ‘qc’	‘c’, L, ‘qc’	‘ ’, R, ‘qc1’
qc1	‘a’, R, ‘qc1’	‘b’, R, ‘qc1’	‘c’, L, ‘qcc’	
qcc	‘c’, N, ‘qT’	‘c’, N, ‘qT’		‘c’, N, ‘qT’

Описание состояний:

- q0 – ищет первый встретившийся символ ‘a’.
- q1 – определяет какой символ стоит после ‘a’.
- qa, qb, qc – переводит курсор на самый первый символ в строке
- qa1, qb1, qc1 – ищет первый символ ‘c’
- qac, qbc, qcc - заменяет в исходной строке символ, предшествующий первому встретившемуся символу ‘c' на символ, следующий за первым встретившимся символом ‘a'
-

Принцип работы машины Тьюринга:

- memory – введённая строка
- index – индекс ячейки (начальное значение 0)
- state – текущее состояние (начальное значение q0)
- stable – таблица состояний

- С помощью цикла while и словаря stable строка преобразуется согласно условию.

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
1.	cccab	bcccab	OK
2.	bbbbcbcbbcca	bbbacbcbbcca	OK

Выводы

В данной лабораторной работе был изучен и применён на практике принцип работы машины Тьюринга.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: app-002.py

```
table = {
    'q0': {'a': ('a', 1, 'q1'), 'b': ('b', 1, 'q0'), 'c': ('c', 1, 'q0'), ' ': (' ', 1, 'q0')},
    'q1': {'a': ('a', -1, 'qa'), 'b': ('b', -1, 'qb'), 'c': ('c', -1, 'qc'), ' ': (' ', -1, 'qa')},

    'qa':{'a': ('a', -1, 'qa'), 'b': ('b', -1, 'qa'), 'c': ('c', -1, 'qa'), ' ': (' ', 1, 'qa1')},
    'qa1':{'a': ('a', 1, 'qa1'), 'b': ('b', 1, 'qa1'), 'c': ('c', -1, 'qac')},
    'qac':{'a': ('a', 0, 'qT'), 'b': ('a', 0, 'qT'), ' ':('a',0,'qT')},

    'qb':{'a': ('a', -1, 'qb'), 'b': ('b', -1, 'qb'), 'c': ('c', -1, 'qb'), ' ': (' ', 1, 'qb1')},
    'qb1':{'a': ('a', 1, 'qb1'), 'b': ('b', 1, 'qb1'), 'c': ('c', -1, 'qbc')},
    'qbc':{'a': ('b', 0, 'qT'), 'b': ('b', 0, 'qT'), ' ':('b',0,'qT')},

    'qc': {'a': ('a', -1, 'qc'), 'b': ('b', -1, 'qc'), 'c': ('c', -1, 'qc'), ' ': (' ', 1, 'qc1')},
    'qc1': {'a': ('a', 1, 'qc1'), 'b': ('b', 1, 'qc1'), 'c': ('c', -1, 'qcc')},
    'qcc': {'a': ('c', 0, 'qT'), 'b': ('c', 0, 'qT'), ' ':('c',0,'qT')},
}

memory = input()
memory = list(' ' + memory + ' ')
index = 0
state = 'q0'

while state != 'qT':
    sim = memory[index]
    new_sim, delta, state = table[state][sim]
    memory[index] = new_sim
```

```
    index += delta
print(''.join(memory))
```