

**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. 3342

Лапшов К.Н.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

## **Цель работы**

Целью данной работы является разработка машины Тьюринга с применением языка программирования 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 'с' на символ, следующий за первым встретившимся символом 'а'. Если первый встретившийся символ 'а' 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.

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

Алфавит ленты в данной работе состоит из символов 'a', 'b', 'c' и ' '.

Программа должна заменить символ слева от первого символа «с» на символ справа от первого символа «а». На основе этого была составлена таблица состояний №1.

Таблица 1: Таблица состояний №1

| Состояние | 'a'          | 'b'          | 'c'           | ' ' (пробел) |
|-----------|--------------|--------------|---------------|--------------|
| q0        | 'a', R, 'q2' | 'b', R, 'q1' | 'c', R, 'q1'  | ' ', R, 'q0' |
| q1        | 'a', R, 'q2' | 'b', R, 'q1' | 'c', R, 'q1'  | ' ', N, 'qE' |
| q2        | 'a', L, 'q3' | 'b', L, 'q4' | 'c', L, 'q5'  | ' ', L, 'q3' |
| q3        | 'a', L, 'q3' | 'b', L, 'q3' | 'c', L, 'q3'  | ' ', R, 'q6' |
| q4        | 'a', L, 'q4' | 'b', L, 'q4' | 'c', L, 'q4'  | ' ', R, 'q7' |
| q5        | 'a', L, 'q5' | 'b', L, 'q5' | 'c', L, 'q5'  | ' ', R, 'q8' |
| q6        | 'a', R, 'q6' | 'b', R, 'q6' | 'c', L, 'q9'  | ' ', N, 'qE' |
| q7        | 'a', R, 'q7' | 'b', R, 'q7' | 'c', L, 'q10' | ' ', N, 'qE' |
| q8        | 'a', R, 'q8' | 'b', R, 'q8' | 'c', L, 'q11' | ' ', N, 'qE' |
| q9        | 'a', N, 'qE' | 'a', N, 'qE' | 'a', N, 'qE'  | 'a', N, 'qE' |
| q10       | 'b', N, 'qE' | 'b', N, 'qE' | 'b', N, 'qE'  | 'b', N, 'qE' |
| q11       | 'c', N, 'qE' | 'c', N, 'qE' | 'c', N, 'qE'  | 'c', N, 'qE' |

q0 — состояние, когда «управляющее устройство» еще не на строке.

q1 — когда «управляющее устройство» на строке, но еще не найден символ «а», из него сразу возможен переход в состояние qE, если не найдется «а» или «с»

q2 — когда символ «а» найден

q3 — состояние, нужное для запоминания символа «а» (когда после первого «а» стоит «а» или пробел). Пока автомат не встретит пробел перед строкой, он будет смещаться влево.

q4 — состояние, нужное для запоминания символа «b» (когда после первого «a» стоит «b»). Пока автомат не встретит пробел перед строкой, он будет смещаться влево.

q5 — состояние, нужное для запоминания символа «c» (когда после первого «a» стоит «c»). Пока автомат не встретит пробел перед строкой, он будет смещаться влево.

q6 — состояние, в которое переходит управляющее устройство после нахождения пробела перед строкой, а также состояние, в котором оно ищет символ «c». Это состояние связано с символом «a»

q7 — состояние, в которое переходит управляющее устройство после нахождения пробела перед строкой, а также состояние, в котором оно ищет символ «c», связано с символом «b»

q8 — состояние, в которое переходит управляющее устройство после нахождения пробела перед строкой, а также состояние, в котором оно ищет символ «c», связано с символом «c»

q9 — найден символ «c», автомат делает шаг назад (влево), заменяет символ перед «c» на символ «a»

q10 — найден символ «c», автомат делает шаг назад (влево), заменяет символ перед «c» на символ «b»

q11 — найден символ «c», автомат делает шаг назад (влево), заменяет символ перед «c» на символ «c»

Создаются переменные для шагов R, L, N. Пользователь вводит строку, начинающуюся и заканчивающуюся минимум одним пробелом. Начальное состояние — q0 (переменная q), позиция управляющего устройства (index) — 0, то есть начало введенной строки.

В цикле while, пока q не равно конечному состоянию qE, берутся из словаря состояний table символ symbol, шаг direction и новое состояние state. Автомат сдвигается на direction, то есть на 0, -1 или 1. Состояние изменяется.

Когда достигнуто конечное состояние qE, печатается измененная лента памяти.

Разработанный программный код см. в приложении А.

## **Выводы**

Были изучены принципы алгоритма работы Машины Тьюринга, а также выполнено практическое задание с ее использованием.



## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

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

table = {
    'q0': {'a': ['a', R, 'q2'], 'b': ['b', R, 'q1'], 'c': ['c', R, 'q1'],
    ' ': [' ', R, 'q0']},
    'q1': {'a': ['a', R, 'q2'], 'b': ['b', R, 'q1'], 'c': ['c', R, 'q1'],
    ' ': [' ', N, 'qE']},
    '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, 'q6']},
    'q4': {'a': ['a', L, 'q4'], 'b': ['b', L, 'q4'], 'c': ['c', L, 'q4'],
    ' ': [' ', R, 'q7']},
    'q5': {'a': ['a', L, 'q5'], 'b': ['b', L, 'q5'], 'c': ['c', L, 'q5'],
    ' ': [' ', R, 'q8']},
    'q6': {'a': ['a', R, 'q6'], 'b': ['b', R, 'q6'], 'c': ['c', L, 'q9'],
    ' ': [' ', N, 'qE']},
    'q7': {'a': ['a', R, 'q7'], 'b': ['b', R, 'q7'], 'c': ['c', L, 'q10'],
    ' ': [' ', N, 'qE']},
    'q8': {'a': ['a', R, 'q8'], 'b': ['b', R, 'q8'], 'c': ['c', L, 'q11'],
    ' ': [' ', N, 'qE']},
    'q9': {'a': ['a', N, 'qE'], 'b': ['a', N, 'qE'], 'c': ['a', N, 'qE'],
    ' ': ['a', N, 'qE']},
    'q10': {'a': ['b', N, 'qE'], 'b': ['b', N, 'qE'], 'c': ['b', N, 'qE'],
    ' ': ['b', N, 'qE']},
    'q11': {'a': ['c', N, 'qE'], 'b': ['c', N, 'qE'], 'c': ['c', N, 'qE'],
    ' ': ['c', N, 'qE']},
}

tape = list(input())
q = 'q0'
index = 0

while q != 'qE':
    symbol, direction, state = table[q][tape[index]]
    tape[index] = symbol
    index += direction
    q = state

print(''.join(tape))
```