

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Информатика»**  
**Тема: Машина Тьюринга**

Студент гр. 3341

Костромитин М.М.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

## **Цель работы**

Целью работы является изучение работы машины Тьюринга, и ее модуляция на языке Python, а также выполнение практического задания с ее помощью.

## 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}.

асаабсбабаасаb

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.

Для примера выше лента будет выглядеть так:

асаабсбабааbab

Алфавит:

- a
- b
- c
- " " (пробел)

Соглашения:

1. Направление движения автомата может быть одно из R (направо), L (налево), N (неподвижно).
2. Гарантируется, что длина 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

1.  $q_1$  – начальное состояние, которое идет вправо по ленте пока не встретится символ  $a$ ,  $b$ ,  $c$ .
2.  $q_2$  – проход по ленте вправо, пока не встретится символ пробела (символ конца строки).
3.  $q_3$  – проход по ленте влево, пока не встретится символ  $a$ , переходит в состояние  $q_4$ .
4.  $q_4$  – проверка, является ли символ после предыдущего ( $a$ ) символом  $a$ , если да, то переход в состояние  $q_5$ , иначе переход в состояние  $q_3$ .
5.  $q_5$  – в зависимости от символа стоящего после двух  $a$ , переходит в состояния  $q_6$ ,  $q_7$  и  $q_8$  соответственно символам  $a$ ,  $b$  и  $c$ .
6.  $q_6$  – проход вправо по ленте, чтобы пройти два подряд идущих символа  $a$ , после этого переход в состояние  $q_9$ .
7.  $q_7$  – то же самое что и  $q_6$ .
8.  $q_8$  – то же самое что и  $q_7$ .
9.  $q_9$  – заменяет символ после двух  $a$  символом  $a$ , переходит в состояние  $q_{12}$ .
10.  $q_{10}$  – заменяет символ после двух  $a$  символом  $b$ , переходит в состояние  $q_{12}$ .
11.  $q_{11}$  – заменяет символ после двух  $a$  символом  $c$ , переходит в состояние  $q_{12}$ .
12.  $q_{12}$  – проходит по ленте вправо пока не встретить пробел, после этого переходит в состояние  $q_n$ .

Описание шагов работы программы:

- Создается словарь `statusDict`, который представляет собой таблицу переходов для каждого состояния и символа на ленте. Каждая ячейка таблицы содержит информацию о новом символе, направлении движения и новом состоянии после перехода.

- Создается словарь `moves`, определяющий значения смещения для каждого из трех возможных направлений движения: 'R' (вправо), 'L' (влево) и 'N' (без смещения).
- Вводится строка пользователя, представляющая собой входные данные для машины Тьюринга.
- Лента инициализируется символом " " (пробел) в начале и в конце, чтобы обеспечить граничные условия для работы машины.
- Запускается цикл `while`, который продолжается до тех пор, пока текущее состояние `state` не станет равным конечному состоянию `qf`.
- На каждом шаге программа извлекает информацию о переходе из таблицы `statusDict` в соответствии с текущим состоянием и символом на ленте.
- Символ на ленте обновляется, текущая позиция смещается в соответствии с указанным направлением, и состояние обновляется на новое.
- Процесс повторяется до достижения конечного состояния `qf`.
- После завершения работы машины Тьюринга выводится результат на экран. Результат представляет собой обработанную часть ленты машины, оставшуюся между первым и текущим символами.

## Тестирование

Результаты тестирования представлены в Таблице 1.

Таблица 1.

№ п/п	Входные данные	Выходные данные	Комментарии
1	abcbacbacbabcaababcba	abcbacbacbabсаасabcba	‘аа’ в середине ленты
2	abcbacbacbacbaa	abcbacbacbacbaab	‘аа’ в конце ленты

## **Выводы**

Лабораторная работа успешно демонстрирует возможность реализации машины Тьюринга с использованием языка программирования Python. Полученный результат соответствует ожидаемому функционалу, что подтверждает корректность работы программы. Реализация машины Тьюринга может быть эффективным инструментом для решения различных задач обработки символьных данных.



## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Исходный файл: main.py

```
statusDict = {
    'q1': {'a': 'a,N,q2', 'b': 'b,N,q2', 'c': 'c,N,q2', ' ': ' ',
    'R,q1'},
    'q2': {'a': 'a,R,q2', 'b': 'b,R,q2', 'c': 'c,R,q2', ' ': ' ',
    'L,q3'},
    'q3': {'a': 'a,L,q4', 'b': 'b,L,q3', 'c': 'c,L,q3', ' ': ' ',
    'N,qn'},
    'q4': {'a': 'a,L,q5', 'b': 'b,L,q3', 'c': 'c,L,q3', ' ': ' ',
    'N,qn'},
    'q5': {'a': 'a,R,q6', 'b': 'b,R,q7', 'c': 'c,R,q8', ' ': ' ',
    'n,qn'},
    'q6': {'a': 'a,R,q6', 'b': 'b,N,q9', 'c': 'c,N,q9', ' ': ' ',
    'N,q9'},
    'q7': {'a': 'a,R,q7', 'b': 'b,N,q10', 'c': 'c,N,q10', ' ': ' ',
    'N,q10'},
    'q8': {'a': 'a,R,q8', 'b': 'b,N,q11', 'c': 'c,N,q11', ' ': ' ',
    'N,q11'},
    'q9': {'a': 'a,N,q12', 'b': 'a,N,q12', 'c': 'a,N,q12', ' ': ' ',
    'a,N,q12'},
    'q10': {'a': 'b,N,q12', 'b': 'b,N,q12', 'c': 'b,N,q12', ' ': ' ',
    'b,N,q12'},
    'q11': {'a': 'c,N,q12', 'b': 'c,N,q12', 'c': 'c,N,q12', ' ': ' ',
    'c,N,q12'},
    'q12': {'a': 'a,R,q12', 'b': 'b,R,q12', 'c': 'c,R,q12', ' ': ' ',
    'N,qn'}
}

moves = {'R': 1, 'L': -1, 'N': 0}

def turingMachine(lenta):
    state = 'q1'
    current = 0
    while state != 'qn':
        symbol, move, newState = statusDict[state]
        [lenta[current]].split(',')
        lenta[current] = symbol
        current += moves[move]
        state = newState
    print(''.join(lenta[1:current]))

lenta = [' '] + list(input()) + [' '] + [' ']
turingMachine(lenta)
```