

**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 и конечные автоматы**

Студент гр. 3344

Волохов М.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

**Цель работы**

Получить навыки составления таблицы машины Тьюринга и работы с машиной Тьюринга.

### **Задание.**

Вариант 4. 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}, которая начинается с символа 'a'.

Напишите программу, которая оборачивает исходную строку. Результат работы алгоритма - исходная последовательность символов в обратном порядке.

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:

Алфавит (можно расширять при необходимости):

- 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 13.
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

Выполнение работы.

Составлена таблица состояний. После идёт цикл, который останавливается при достижении машиной конечного состояния. В переменные state, pos, symb записываются текущее состояние, позиция и символ, который нужно записать в ленту, соответственно.

	a	b	c	D	« »
q1	a;N;q2				« »;R;q1
q2	D;L;q3	D;L;q4	D;L;q5	D;R;q2	« »;L;q7
q3	a;L;q3	b;L;q3	c;L;q3	D;L;q3	a;R;q6
q4	a;L;q4	b;L;q4	c;L;q4	D;L;q4	b;R;q6
q5	a;L;q5	b;L;q5	c;L;q5	D;L;q5	c;R;q6
q6	a;R;q6	b;R;q6	c;R;q6	D;R;q2	
q7	a;N;q8			« »;L;q7	

q1 — Начальное состояние, для нахождения начала строки

q2 — Заменяет найденный символ на временный «D» и переходит к состоянию, которое соответствует символу: a – q3, b – q4, c – q5; если видит пробел, то переходит к состоянию q7

q3 — Проходит влево и вставляет «a» на место первого найденного пробела

q4 — Проходит влево и вставляет «b» на место первого найденного пробела

q5 — Проходит влево и вставляет «c» на место первого найденного пробела

q6 — Проходит вправо до первого символа «D»

q7 — Удаляет символы «D»

q8 — Конечное состояние

### Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abcabc	cbacba	-
2.	cabacbbcc	ccbbscabac	-

## **Выводы**

Были освоены принципы работы машины Тьюринга. Был написан алгоритм для подражания машине Тьюринга, инвертирующий входную строку.

# **ПРИЛОЖЕНИЕ А** **ИСХОДНЫЙ КОД ПРОГРАММЫ**

Название файла: Volokhov\_Mikhail\_lb3.py

```
# правила машины Тьюринга в виде таблицы
table = {
    'q1': {'a': ('a', 0, 'q2'), 'b': ('b', 1, 'q1')},
    'q2': {'a': ('D', -1, 'q3'), 'b': ('D', -1, 'q4'), 'c': ('D', -1, 'q5')},
    'q3': {'a': ('a', -1, 'q3'), 'b': ('b', -1, 'q3'), 'c': ('c', -1, 'q7')},
    'q4': {'a': ('a', -1, 'q4'), 'b': ('b', -1, 'q4'), 'c': ('c', -1, 'q6')},
    'q5': {'a': ('a', -1, 'q5'), 'b': ('b', -1, 'q5'), 'c': ('c', -1, 'q6')},
    'q6': {'a': ('a', 1, 'q6'), 'b': ('b', 1, 'q6'), 'c': ('c', 1, 'q6')},
    'q7': {'a': ('a', 0, 'q8'), 'b': ('b', 1, 'q7')},
    'q8': {'a': ('a', 0, 'q8'), 'b': ('b', 1, 'q7')},
}

# Инициализация ленты с пробелами и ввод пользователя
strip = [' ' * 15]
strip += [x for x in input()]
strip += [' ' * 15]

# Начальные значения
state = 'q1'
pos = 0

# Запуск машины Тьюринга
while state != 'q8':
    symb, step, state = table[state][strip[pos]]
    # Обновление ленты и позиции в соответствии с правилами
    strip[pos] = symb
    pos += step

print(''.join(strip))
```