

**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. 2381

Соколов С.А.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2022

**Цель работы.**

Изучить машину 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.

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 длинна 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.

Таблица 1 – Состояния машины Тьюринга

Состояния	a	b	c	" "	Описание
q0	a R q1	" " R q2	" " R q3	" " R q0	Состояние до начала обработки строки
q1	a R q1	b R q2	c R q3	" " R q7	Получение последней буквы
q2	a R q1	b R q2	c R q3	" " R q8	
q3	a R q1	b R q2	c R q3	" " R q9	
q4	a L q4	b L q4	c L q4	" " L q5	Перемещение к началу второго слова
q5	" " L q6	" " L q6	" " L q6	" " L q5	Возврат на первое слово и удаление последней буквы
q6	a R q10	b R q11	c R q12	" " N q13	Перенаправление на состояние перемещения к новому слову
q7	a R q7	b R q7	c R q7	a L q4	Запись буквы в новое слово
q8	a R q8	b R q8	c R q8	b L q4	
q9	a R q9	b R q9	c R q9	c L q4	
q10	a R q7	b R q7	c R q7	" " R q10	Перемещение к концу второго слова
q11	a R q8	b R q8	c R q8	" " R q11	
q12	a R q9	b R q9	c R q9	" " R q12	

q13 – конечное состояние.

Переменные *R*, *N*, *L* хранят значение сдвига относительно текущей позиции.

Переменная *alphabet* хранит словарь соотносящий символ и его номер столбца в таблице состояний.

Переменная *states* хранит словарь состояний машины Тьюринга. Значения, получаемые по ключу, хранят: символы, которые необходимо записать на место текущего, перемещение относительно текущей позиции, следующее состояние.

Переменная *tape* хранит состояние ленты, *state* текущее состояние автомата, *index* – позицию на ленте, в цикле эмулируется изменение состояний машины Тьюринга.

Разработанный программный код см. в приложении А.

Результаты тестирования см. в приложении Б.

**Выводы.**

Была изучена машина Тьюринга.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

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

alphabet = {
    'a': 0, 'b': 1, 'c': 2, ' ': 3
}

states = {
    # Начальное состояние
    'q0': {
        'write': ['a', ' ', ' ', ' '],
        'move': [R, R, R, R],
        'nextState': ['q1', 'q2', 'q3', 'q0']
    },

    # Получение последней буквы
    'q1': {
        'write': ['a', 'b', 'c', ' '],
        'move': [R, R, R, R],
        'nextState': ['q1', 'q2', 'q3', 'q7']
    },
    'q2': {
        'write': ['a', 'b', 'c', ' '],
        'move': [R, R, R, R],
        'nextState': ['q1', 'q2', 'q3', 'q8']
    },
    'q3': {
        'write': ['a', 'b', 'c', ' '],
        'move': [R, R, R, R],
        'nextState': ['q1', 'q2', 'q3', 'q9']
    },

    # Перемещение к началу второго слова
    'q4': {
        'write': ['a', 'b', 'c', ' '],
        'move': [L, L, L, L],
        'nextState': ['q4', 'q4', 'q4', 'q5']
    },

    # Возврат на первое слово и удаление последней буквы
    'q5': {
        'write': [' ', ' ', ' ', ' '],
        'move': [L, L, L, L],
        'nextState': ['q6', 'q6', 'q6', 'q5']
    },

    # Перенаправление на путь к новому слову
    'q6': {
        'write': ['a', 'b', 'c', ' '],
        'move': [R, R, R, N],
        'nextState': ['q10', 'q11', 'q12', 'q13']
    },

    # Запись буквы в новое слово
    'q7': {
        'write': ['a', 'b', 'c', 'a'],
```

```

        'move': [R, R, R, L],
        'nextState': ['q7', 'q7', 'q7', 'q4']
    },
    'q8': {
        'write': ['a', 'b', 'c', 'b'],
        'move': [R, R, R, L],
        'nextState': ['q8', 'q8', 'q8', 'q4']
    },
    'q9': {
        'write': ['a', 'b', 'c', 'c'],
        'move': [R, R, R, L],
        'nextState': ['q9', 'q9', 'q9', 'q4']
    },
    # Перемещение к концу второго слова
    'q10': {
        'write': ['a', 'b', 'c', ' '],
        'move': [R, R, R, R],
        'nextState': ['q7', 'q7', 'q7', 'q10']
    },
    'q11': {
        'write': ['a', 'b', 'c', ' '],
        'move': [R, R, R, R],
        'nextState': ['q8', 'q8', 'q8', 'q11']
    },
    'q12': {
        'write': ['a', 'b', 'c', ' '],
        'move': [R, R, R, R],
        'nextState': ['q9', 'q9', 'q9', 'q12']
    },
}

if __name__ == '__main__':
    tape = list(input())
    tape += ((len(tape) + 2) * ' ')

    state = 'q0'
    index = 0
    while state != 'q13':
        oldValue = tape[index]

        tape[index] = states[state]['write'][alphabet[oldValue]]
        index += states[state]['move'][alphabet[oldValue]]
        state = states[state]['nextState'][alphabet[oldValue]]

    print(''.join(tape))

```

**ПРИЛОЖЕНИЕ Б**  
**ТЕСТИРОВАНИЕ**

Таблица Б.1 - Примеры тестовых случаев

№ п/п	Входные данные	Выходные данные
1.	bacbc	cbca
2.	abcabc	cbacba
3.	abacbbc	cbbcaba
4.	babcabcbacbabacab	bacababcabcbacbacba