

**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(ка) гр. 0000

Шляхтин М.Д.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2022

### **Цель работы.**

Изучить принцип работы машины Тьюринга, осуществить алгоритм для решения задачи лабораторной работы.

### **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 — Программа для машины Тьюринга

Состояние	a	b	c	« »(пробел)	x
q1	a,R,q2	b,R,q2	c,R,q2	« »,R,q1	-
q2	a,R,q2	b,R,q2	c,R,q2	« »,L,q3	-
q3	a,L,q4	b,L,q4	c,L,q4	-	-
q4	x,R,q5	x,R,q6	x,R,q7	-	-
q5	a,R,q9	b,R,q9	c,R,q9	-	x,R,q5
q6	a,R,q10	b,R,q10	c,R,q10	-	x,R,q6
q7	a,R,q11	b,R,q11	c,R,q11	-	x,R,q7
q8	a,L,q8	b,L,q8	c,L,q8	-	x,L,q12
q9	a,R,q9	b,R,q9	c,R,q9	a,R,q9	-
q10	a,R,q10	b,R,q10	c,R,q10	b,R,q10	-
q11	a,R,q11	b,R,q11	c,R,q11	c,R,q11	x,L,q12
q12	x,R,q5	x,R,q6	x,R,q7	« »,R,q13	x,L,q12
q13	a,N,qT	b,N,qT	c,N,qT	-	« »,R,q13

- «q1» - начальное положение, ищем конец строки.
- «q2» - в этом состоянии отмечаем, что первый пробел был пройден
- «q3» - в данном состоянии мы находимся на последнем символе исходной строки, то есть на первом символе обработанной строки.
- «q4» - определяем, какой символ будет следующим в преобразованной строке, и записываем вместо него «x»
- «q5» - следующий символ, который необходимо записать это «a», идем по строке вправо
- «q6» - следующий символ, который необходимо записать это «b», идем по строке вправо
- «q7» - следующий символ, который необходимо записать это «c», идем по строке вправо
- «q8» - в этом состоянии осуществляется поиск следующего символа, который нужно записать в обернутую строку

- «q9» - доходим до конца строки и записываем символ «a» в конец преобразованной строки
- «q10» - доходим до конца строки и записываем символ «b» в конец преобразованной строки
- «q11» - доходим до конца строки и записываем символ «c» в конец преобразованной строки
- «q12» - записываем вместо символа исходной строки «x», чтобы отметить, что он будет добавлен в конец измененной строки
- «q13» - все элементы исходной строки были записаны в обратном порядке; вместо символов «x» записываем пробелы
- «qT» - конечное состояние, работы программы завершена.

Переменные R, L, N используются для осуществления сдвига (на 1 вправо, на 1 влево и на 0 соответственно). В переменной memory хранится исходная строка. С помощью переменной buf ее размер увеличивается, чтобы рядос можно было записать преобразованную строку. Переменная state хранит в себе состояние, в котором находится машина Тьюринга, переменная index используется для перемещения по строке(т.е. по элементам memory). Цикл while работает пока текущее состояние программы не будет равно qT. В переменную current\_symbol записывается текущий символ строки. Переменная future получает из таблицы новое состояние, шаг и символ для записи. Далее значения передаются соответствующим переменным.

Разработанный код программы находится в приложении А.

### **Тестирование.**

Результаты тестирования смотреть в приложении Б.

### **Выводы.**

Был изучен принцип работы машины Тьюринга, а также реализован алгоритм решения задачи по построению строки обратной исходной. Для этого

была создана программа на языке Python, имитирующая механизм работы машины Тьюринга.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```

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

table = {'q1': {'a': ['a', R, 'q2'], 'b': ['b', R, 'q2'], 'c':
['c', R, '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, 'q4'], 'c':
['c', L, 'q4']}},
        'q4': {'a': ['x', R, 'q5'], 'b': ['x', R, 'q6'], 'c':
['x', R, 'q7']}},
        'q5': {'a': ['a', R, 'q9'], 'b': ['b', R, 'q9'], 'c':
['c', R, 'q9'], 'x': ['x', R, 'q5']}},
        'q6': {'a': ['a', R, 'q10'], 'b': ['b', R, 'q10'], 'c':
['c', R, 'q10'], 'x': ['x', R, 'q6']}},
        'q7': {'a': ['a', R, 'q11'], 'b': ['b', R, 'q11'], 'c':
['c', R, 'q11'], 'x': ['x', R, 'q7']}},
        'q8': {'a': ['a', L, 'q8'], 'b': ['b', L, 'q8'], 'c':
['c', L, 'q8'], 'x': ['x', L, 'q12']}},
        'q9': {'a': ['a', R, 'q9'], 'b': ['b', R, 'q9'], 'c':
['c', R, 'q9'], ' ': ['a', L, 'q8']}},
        'q10': {'a': ['a', R, 'q10'], 'b': ['b', R, 'q10'], 'c':
['c', R, 'q10'], ' ': ['b', L, 'q8']}},
        'q11': {'a': ['a', R, 'q11'], 'b': ['b', R, 'q11'], 'c':
['c', R, 'q11'], ' ': ['c', L, 'q8']}},
        'q12': {'a': ['x', R, 'q5'], 'b': ['x', R, 'q6'], 'c':
['x', R, 'q7'], 'x': ['x', L, 'q12'], ' ': [' ', R, 'q13']}},
        'q13': {'x': [' ', R, 'q13'], 'a': ['a', N, 'qT'], 'b':
['b', N, 'qT'], 'c': ['c', N, 'qT']}}

}

memory = list(input())
buf = [' ' for i in range(30)]
memory = buf + memory + buf
state = 'q1'
index = 0
states = [state]
while state != 'qT':

    current_symbol = memory[index]
    future = table[state][current_symbol]
    memory[index] = future[0]
    index += future[1]
    state = future[2]
    states += [state]
    print(*memory)

print(*memory, sep='')

```

**ПРИЛОЖЕНИЕ Б**  
**ТЕСТИРОВАНИЕ**

Таблица 2 - Примеры тестовых случаев

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abcabc	cbacba	-
2.	abacbbc	cbbcaba	-
3.	abcabcabcabcc	ccbacbacbacba	-