

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

Рыжиков И.С.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2022

Цель работы

Изучить принципы работы машины Тьюринга. Написать собственный автомат для выполнения задания.

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}.

			a	c	a	a	b	c	b	a	b	a	a	c	a	b			
--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

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:

			a	c	a	a	b	c	b	a	b	a	a	b	a	b			
--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

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.

В отчете предоставьте таблицу состояний. Отдельно кратко опишите каждое состояние, например:

q1 - начальное состояние, которое необходимо, чтобы найти первый встретившийся символ 'a'.

Пример:

Test	Input	Result
#1	abcaabc	abcaacc
#2	aabbaa	aabbaab

Выполнение работы

Описание состояний

- q_0 — конечное состояние;
- q_1 — поиск начала слова;
- q_2 — переход в конец слова;
- q_3 — найти один символ 'a';
- q_4 — найти последние два подряд идущих символа 'a';
- q_5 — определить символ, на которой требуется заменить;
- q_6 и q_7 , q_9 и q_{10} , q_{12} и q_{13} — перейти на два шага вперед, сохраняя информацию о заменяющем символе;
- q_8 , q_{11} , q_{14} — заменить символ на 'a', 'b' или 'c' соответственно.

Таблица 1. Состояния

Состояние	'a'	'b'	'c'	' ' (пробел)
q_1	'a', R, ' q_2 '	'b', R, ' q_2 '	'c', R, ' q_2 '	' ', R, ' q_1 '
q_2	'a', R, ' q_2 '	'b', R, ' q_2 '	'c', R, ' q_2 '	' ', L, ' q_3 '
q_3	'a', L, ' q_4 '	'b', L, ' q_3 '	'c', L, ' q_3 '	
q_4	'a', L, ' q_5 '	'b', L, ' q_3 '	'c', L, ' q_3 '	
q_5	'a', R, ' q_6 '	'b', R, ' q_9 '	'c', R, ' q_{12} '	
q_6	'a', R, ' q_7 '			
q_7	'a', R, ' q_8 '			
q_8	'a', N, ' q_0 '	'a', N, ' q_0 '	'a', N, ' q_0 '	'a', N, ' q_0 '
q_9	'a', R, ' q_{10} '			
q_{10}	'a', R, ' q_{11} '			
q_{11}	'b', N, ' q_0 '	'b', N, ' q_0 '	'b', N, ' q_0 '	'b', N, ' q_0 '
q_{12}	'a', R, ' q_{13} '			
q_{13}	'a', R, ' q_{14} '			
q_{14}	'c', N, ' q_0 '	'c', N, ' q_0 '	'c', N, ' q_0 '	'c', N, ' q_0 '

Принцип работы программы

Сохраняем введенную из консоли строку в переменную *tape* в виде списка символов. Передаем её в функцию *turing*, которая в соответствии с таблицей состояний выполняет программу. Для чего в ней локально создается переменная *state*, хранящее текущее состояние машины, *i* — индекс текущего символа в списке, и *move* — шаг. Запускаем внутри функции цикл до тех пор, пока состояние машины не будет равно конечному. Записываем новый символ на ленту, делаем шаг и обновляем текущее состояние системы. После выполнения функции *turing*, которая имитирует работу машины Тьюринга, выводим получившуюся строку на экран.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	' аассааб '	' аассаас '	OK
2.	' абсааabc '	' абсааacc '	OK
3.	' ааббаа '	' ааббааб '	OK
4.	' асбаа '	' асбааб '	OK

Выводы

Была изучена машина Тьюринга. Создана функция, которая заменяет в исходной строке символ, идущий после последних двух встретившихся символов 'а', на предшествующий им символ. Разработана программа на языке Python, имитирующая работу машины Тьюринга.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: *main.py*

```
def turing(tape):
    table = {
        # Find a word
        'q1': {
            ' ': [' ', 1, 'q1'],
            'a': ['a', 1, 'q2'],
            'b': ['b', 1, 'q2'],
            'c': ['c', 1, 'q2']
        },
        # Go to the end of word
        'q2': {
            ' ': [' ', -1, 'q3'],
            'a': ['a', 1, 'q2'],
            'b': ['b', 1, 'q2'],
            'c': ['c', 1, 'q2']
        },
        # Find one 'a'
        'q3': {
            'a': ['a', -1, 'q4'],
            'b': ['b', -1, 'q3'],
            'c': ['c', -1, 'q3']
        },
        # Find the last couple of 'a' in a row
        'q4': {
            'a': ['a', -1, 'q5'],
            'b': ['b', -1, 'q3'],
            'c': ['c', -1, 'q3']
        },
        # Branching: Define a replacement character
        'q5': {
            'a': ['a', 1, 'q6'],
            'b': ['b', 1, 'q9'],
            'c': ['c', 1, 'q12']
        },
        # Branch of 'a':
        # Go back
        'q6': {
            'a': ['a', 1, 'q7']
        },
        'q7': {
            'a': ['a', 1, 'q8']
        },
        # Replace with 'a'
        'q8': {
            'a': ['a', 0, 'q0'],
            'b': ['a', 0, 'q0'],
            'c': ['a', 0, 'q0'],
            ' ': ['a', 0, 'q0']
        }
    }
```

```

    },
    # Branch of 'b':
    # Go back
    'q9': {
        'a': ['a', 1, 'q10']
    },
    'q10': {
        'a': ['a', 1, 'q11']
    },
    # Replace with 'b'
    'q11': {
        'a': ['b', 0, 'q0'],
        'b': ['b', 0, 'q0'],
        'c': ['b', 0, 'q0'],
        ' ': ['b', 0, 'q0']
    },
    # Branch of 'c':
    # Go back
    'q12': {
        'a': ['a', 1, 'q13']
    },
    'q13': {
        'a': ['a', 1, 'q14']
    },
    # Replace with 'c'
    'q14': {
        'a': ['c', 0, 'q0'],
        'b': ['c', 0, 'q0'],
        'c': ['c', 0, 'q0'],
        ' ': ['c', 0, 'q0']
    },
    },
}
state = 'q1'
i = 0
while state != 'q0':
    tape[i], move, state = table[state][tape[i]]
    i += move

if __name__ == '__main__':
    tape = list(input())
    turing(tape)
    print(''.join(tape))

```