

**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. 3341

Шуменков А.П.

Преподаватель

Иванов Д. В.

Санкт-Петербург

2023

Цель работы

Целью лабораторной работы является изучение работы конечных автоматов, в частности машины Тьюринга.

Для достижения поставленной цели требуется решить следующие задачи:

- 1) Ознакомиться с концепцией машины Тьюринга
- 2) Создать программу, моделирующую работу машины Тьюринга, выполняющую определенную задачу.

Задание

Вариант работы №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'.

			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 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.
6. Нельзя использовать дополнительную ленту, в которую записывается результат.

Ваша программа должна вывести полученную ленту после завершения работы.

Выполнение работы

Для решения задачи таблица состояний машины Тьюринга (см. ниже) была реализована в виде словаря .

Далее осуществляется ввод строки с клавиатуры в переменную *lenta* (лента машины). Затем в массив *lenta* добавляется несколько (13+1) пустых ячеек. Дополнительно добавляется одна пустая ячейка в начало массива (перед началом слова согласно заданию).

Переменная *idx* используется для обозначения текущей ячейки, обрабатываемой машиной Тьюринга. (Изначально равна 0). Переменная *sost* содержит текущее состояние машины Тьюринга. Изначально *q_0*.

Далее в цикле *while* перебираются символы ленты (массива *lenta*). Текущий символ записывается в переменную *symbol*. Далее обновляются переменные *new_symbol* (на что меняется текущий символ), *move* (сдвиг ленты машины Тьюринга: 1 — вправо, -1 — влево), *sost* (новое состояние машины Тьюринга).

Функция возвращает строку, полученную из списка *tape* методом *join()*.

Массиву *sostoyaniya* соответствует таблица состояний машины Тьюринга:

	'a'	'b'	'c'	'A'	'B'	'C'	'0'	' '
<i>q_0</i>	a; 1; <i>q_1</i>	b; 1; <i>q_1</i>	c; 1; <i>q_1</i>					' '; 1; <i>q_0</i>
<i>q_1</i>	a; 1; <i>q_1</i>	b; 1; <i>q_1</i>	c; 1; <i>q_1</i>					' '; -1; <i>q_2</i>
<i>q_2</i>	0; 1; <i>q_3</i>	0; 1; <i>q_4</i>	0; 1; <i>q_5</i>	A; -1; <i>q_2</i>	B; -1; <i>q_2</i>	C; -1; <i>q_2</i>	0; -1; <i>q_2</i>	' '; 1; <i>q_0</i>
<i>q_3</i>	a; 1; <i>q_3</i>	b; 1; <i>q_3</i>	c; 1; <i>q_3</i>	A; 1; <i>q_3</i>	B; 1; <i>q_3</i>	C; 1; <i>q_3</i>	0; 0; <i>q_3</i>	A; -1; <i>q_2</i>
<i>q_4</i>	a; 1; <i>q_3</i>	b; 1; <i>q_4</i>	c; 1; <i>q_4</i>	A; 1; <i>q_4</i>	B; 1; <i>q_4</i>	C; 1; <i>q_4</i>	0; 0; <i>q_4</i>	B; -1; <i>q_2</i>

q ₅	a; 1; q ₅	b; 1; q ₅	c; 1; q ₅	A; 1; q ₅	B; 1; q ₅	C; 1; q ₅	0; 0; q ₅	C; -1; q ₂
q ₆				a; 1; q ₆	b; 1; q ₆	c; 1; q ₆	' ' ; 1; q ₆	' ' ; 0; q _T
q _T								

q₀ — начальное состояние. Машина двигается по ленте вправо, пока не найдет первый символ, отличный от пробела (a, b, c). После чего переходит в состояние q₁.

q₁ — машина нашла первый символ строки, после чего двигается вправо до конца строки, пока снова не найдет пробел. Потом переходит в состояние q₂.

q₂ — машина двигается по строке в обратном направлении (влево), пока не встретит пробел. (Т.е. пока не дойдет до начала строки). Если в данном состоянии машина встречает символ отличный от 0, она заменяет его на 0 и переходит в состояние q₃, если символ был a, состояние q₄ — если b, и q₅ — если c. Если символ был A, B или C он не меняется, машина продолжает двигаться влево. Если машина в данном состоянии встречает пробел, машина переходит в состояние q₆.

q₃ — машина двигается вправо, пока не находит первый пробел. После чего она заменяет его на A и переходит в состояние q₂.

q₄ — машина двигается вправо, пока не находит первый пробел. После чего она заменяет его на B и переходит в состояние q₂.

q₅ — машина двигается вправо, пока не находит первый пробел. После чего она заменяет его на C и переходит в состояние q₂.

q₆ — машина двигается вправо, пока не найдет пробел. Она заменяет 0 на пробелы, а строчные буквы на прописные. После чего машина переходит в терминальное состояние.

Разработанный программный код см. в приложении А.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	abcabc	cbacba
2.	cbcbabcb	bcbabcbc
1.	aaccbb	bbccaa

Выводы

Была разработана программа на языке программирования Python, симулирующая работу машины Тьюринга. Была описана программа машины Тьюринга, с помощью которой автомат способен обработать строку символов определённым образом.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
sostoyaniya = {'q_0': {'a': ('a', 1, 'q_1'),
                        'b': ('b', 1, 'q_1'),
                        'c': ('c', 1, 'q_1'),
                        ' ': (' ', 1, 'q_0')
                       },
               'q_1': {'a': ('a', 1, 'q_1'),
                        'b': ('b', 1, 'q_1'),
                        'c': ('c', 1, 'q_1'),
                        ' ': (' ', -1, 'q_2')
                       },
               'q_2': {'a': ('0', 1, 'q_3'),
                        'b': ('0', 1, 'q_4'),
                        'c': ('0', 1, 'q_5'),
                        'A': ('A', -1, 'q_2'),
                        'B': ('B', -1, 'q_2'),
                        'C': ('C', -1, 'q_2'),
                        '0': ('0', -1, 'q_2'),
                        ' ': (' ', 1, 'q_6'),
                       },
               'q_3': {'a': ('a', 1, 'q_3'),
                        'b': ('b', 1, 'q_3'),
                        'c': ('c', 1, 'q_3'),
                        'A': ('A', 1, 'q_3'),
                        'B': ('B', 1, 'q_3'),
                        'C': ('C', 1, 'q_3'),
                        '0': ('0', 1, 'q_3'),
                        ' ': ('A', -1, 'q_2')
                       },
               'q_4': {'a': ('a', 1, 'q_4'),
                        'b': ('b', 1, 'q_4'),
                        'c': ('c', 1, 'q_4'),
                        'A': ('A', 1, 'q_4'),
                        'B': ('B', 1, 'q_4'),
                        'C': ('C', 1, 'q_4'),
                        '0': ('0', 1, 'q_4'),
                        ' ': ('B', -1, 'q_2')
                       },
               'q_5': {'a': ('a', 1, 'q_5'),
                        'b': ('b', 1, 'q_5'),
                        'c': ('c', 1, 'q_5'),
                        'A': ('A', 1, 'q_5'),
                        'B': ('B', 1, 'q_5'),
                        'C': ('C', 1, 'q_5'),
                        '0': ('0', 1, 'q_5'),
                        ' ': ('C', -1, 'q_2')
                       },
               'q_6': {'0': (' ', 1, 'q_6'),
                        'A': ('a', 1, 'q_6'),
                        'B': ('b', 1, 'q_6'),
                        'C': ('c', 1, 'q_6'),
                       }
```

```

        ' ': (' ', 0, 'qT')
    }

    lenta = list(input())
    lenta += [' ' for x in range(14)]
    lenta = [' '] + lenta
    lenta += [' ']
    idx = 0
    sost = 'q_0'

    while sost != 'qT':
        symbol = lenta[idx]

        new_symbol, move, sost = sostoyaniya[sost][symbol]

        lenta[idx] = new_symbol
        idx += move

    print(''.join(lenta).replace(' ', ''))

```