

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. 3343

Пименов П.В.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Изучить принцип работы Машины Тьюринга, научиться создавать для нее программы, создать программу для МТ, выполняющую преобразования над строкой.

Задание

Вариант 1. Написать программу, которая удаляет в исходной строке два символа, следующих за первым встретившимся символом 'b'. Если первый встретившийся символ 'b' – последний в строке, то удалить его. Если первый встретившийся символ 'b' – предпоследний в строке, то удалить один символ, следующий за ним, т. е. последний в строке. Если в строке символ 'b' отсутствует, то удалить самый первый символ строки. После удаления в строке не должно оставаться пробелов и пустых мест. 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: { '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.

Программа должна вывести полученную ленту после завершения работы.

Выполнение работы

Задача 1. Для выполнения поставленной задачи была создана следующая таблица состояний:

	'a'	'b'	'c'	' '
q1	'a'; N; q2	'b'; N; q2	'c'; N; q2	' '; R; q1
q2	'a'; R; q2	'b'; R; q3	'c'; R; q2	' '; L; q16
q3	'a'; R; q4	'b'; R; q4	'c'; R; q4	' '; L; q17
q4	'a'; L; q5	'b'; L; q5	'c'; L; q5	' '; L; q17
q5	'a'; L; q5	'b'; L; q5	'c'; L; q5	' '; R; q6
q6	'a'; R; q6	'b'; R; q7	'c'; R; q6	
q7	' '; R; q8	' '; R; q8	' '; R; q8	' '; R; q8
q8	' '; R; q9	' '; R; q9	' '; R; q9	' '; R; q9
q9	' '; L; q10	' '; L; q12	' '; L; q14	' '; N; q18
q10	'a'; R; q11	'b'; R; q11	'c'; R; q11	' '; L; q10
q11				'a'; R; q7
q12	'a'; R; q13	'b'; R; q13	'c'; R; q13	' '; L; q12
q13				'b'; R; q7
q14	'a'; R; q15	'b'; R; q15	'c'; R; q15	' '; L; q14
q15				'c'; R; q7
q16	'a'; L; q16	'b'; L; q16	'c'; L; q16	' '; R; q17
q17	' '; N; q18	' '; N; q18	' '; N; q18	' '; N; q18

Рисунок 1 – Таблица состояний для Машины Тьюринга

Описание состояний:

- *q1* – начальное состояние, поиск первой буквы
- *q2* – проверка, есть ли 'b' в строке
- *q3* – проверка, последний ли 'b' символ
- *q4* – проверка, предпоследний ли 'b' символ
- *q5* – случай когда 'b' в строке, он не последний и не предпоследний, смещение до первого пробела слева от строки
- *q6* – поиск первого 'b' в строке
- *q7* – замена символа на курсоре на пробел, смещение вправо (фаза 1)
- *q8* – замена символа на курсоре на пробел, смещение вправо (фаза 2)

- *q9* – выбор варианта смещения символа на курсоре в зависимости от его значения ('a', 'b' или 'c')
- *q10* – смещение влево на первый пробел после буквы (для варианта с 'a')
- *q11* – замена символа на курсоре на 'a', смещение вправо и переход в состояние *q7*
- *q12* – смещение влево на первый пробел после буквы (для варианта с 'b')
- *q13* – замена символа на курсоре на 'b', смещение вправо и переход в состояние *q7*
- *q14* – смещение влево на первый пробел после буквы (для варианта с 'c')
- *q15* – замена символа на курсоре на 'c', смещение вправо и переход в состояние *q7*
- *q16* – поиск самого левого непробельного символа в строке
- *q17* – удаление символа на курсоре и переход в терминальное состояние
- *q18* – терминальное состояние, конец программы

Описание принципа работы модели МТ в коде: *tape* – лента МТ, список символов, поданный на вход; *moves* – словарь, представляющий собой модель программы для МТ, его формат: ключ – кортеж из значений «текущее состояние МТ» и «символ на курсоре», значение – кортеж из значений «значение для записи», «смещение» и «новое состояние». Обращаясь к значению словаря по соответствующему ключу, можно получить следующий шаг программы; *last_state* – последнее состояние МТ; *last_index* – последнее положение курсора. Далее, циклом *while*, который работает, пока программа для МТ не перейден в терминальное состояние, вычисляется следующий шаг и реализуется (записывается символ, смещается курсор, меняется состояние). После выполнения цикла выводится состояние ленты МТ.

Разработанный программный код см. в приложении А.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	bbbac	bac	Программа работает корректно
2.	aaacccbc	aaacccb	Программа работает корректно
3.	caaab	caaa	Программа работает корректно
4.	acccaaac	cccaaac	Программа работает корректно

Выводы

Был изучен принцип работы Машины Тьюринга, а также создана программа для МТ, выполняющая преобразования над строкой.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
L = -1
R = 1
N = 0
# (state, value): (value, move, state)
moves = {
    (1, 'a'): ('a', N, 2),
    (1, 'b'): ('b', N, 2),
    (1, 'c'): ('c', N, 2),
    (1, ' '): (' ', R, 1),

    (2, 'a'): ('a', R, 2),
    (2, 'b'): ('b', R, 3),
    (2, 'c'): ('c', R, 2),
    (2, ' '): (' ', L, 16),

    (3, 'a'): ('a', R, 4),
    (3, 'b'): ('b', R, 4),
    (3, 'c'): ('c', R, 4),
    (3, ' '): (' ', L, 17),

    (4, 'a'): ('a', L, 5),
    (4, 'b'): ('b', L, 5),
    (4, 'c'): ('c', L, 5),
    (4, ' '): (' ', L, 17),

    (5, 'a'): ('a', L, 5),
    (5, 'b'): ('b', L, 5),
    (5, 'c'): ('c', L, 5),
    (5, ' '): (' ', R, 6),

    (6, 'a'): ('a', R, 6),
    (6, 'b'): ('b', R, 7),
    (6, 'c'): ('c', R, 6),

    (7, 'a'): (' ', R, 8),
    (7, 'b'): (' ', R, 8),
    (7, 'c'): (' ', R, 8),
    (7, ' '): (' ', R, 8),

    (8, 'a'): (' ', R, 9),
    (8, 'b'): (' ', R, 9),
    (8, 'c'): (' ', R, 9),
    (8, ' '): (' ', R, 9),

    (9, 'a'): (' ', L, 10),
    (9, 'b'): (' ', L, 12),
    (9, 'c'): (' ', L, 14),
    (9, ' '): (' ', N, 18),
```

```

(10, 'a'): ('a', R, 11),
(10, 'b'): ('b', R, 11),
(10, 'c'): ('c', R, 11),
(10, ' '): (' ', L, 10),

(11, ' '): ('a', R, 7),

(12, 'a'): ('a', R, 13),
(12, 'b'): ('b', R, 13),
(12, 'c'): ('c', R, 13),
(12, ' '): (' ', L, 12),

(13, ' '): ('b', R, 7),

(14, 'a'): ('a', R, 15),
(14, 'b'): ('b', R, 15),
(14, 'c'): ('c', R, 15),
(14, ' '): (' ', L, 14),

(15, ' '): ('c', R, 7),

(16, 'a'): ('a', L, 16),
(16, 'b'): ('b', L, 16),
(16, 'c'): ('c', L, 16),
(16, ' '): (' ', R, 17),

(17, 'a'): (' ', N, 18),
(17, 'b'): (' ', N, 18),
(17, 'c'): (' ', N, 18),
(17, ' '): (' ', N, 18)
}
tape = [x for x in input()]
last_state = 1
last_index = 0
while last_state != 18:
    value, move, state = moves[(last_state, tape[last_index])]
    tape[last_index] = value
    last_index += move
    last_state = state
print(''.join(tape))

```