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

Студентка гр. 3342	Смирнова Е.С.
Преподаватель	Иванов Д.В.

Санкт-Петербург 2023

Цель работы

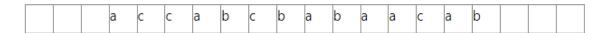
Изучить принцип работы машины Тьюринга и разработать программу для решения поставленной задачи, имитирующую механизм работы данной машины.

Задание

(Вариант 4)

На вход программе подается строка неизвестной длины. Каждый элемент является значением в ячейке памяти ленты Машины Тьюринга.

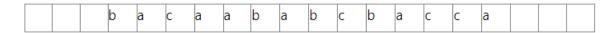
На ленте находится последовательность латинских букв из алфавита {a, b, c}, которая начинается с символа 'a'.



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.

Для примера выше лента будет выглядеть так:



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

Ваша программа должна вывести полученную ленту после завершения работы.

В отчет включите таблицу состояний. Отдельно кратко опишите каждое состояние, например:

q1 - начальное состояние, которое необходимо, чтобы обнаружить конец строки.

Выполнение работы

Таблица состояний представлена в табл.1.

Таблица 1 – Таблица состояний

q	'a'	<i>'b'</i>	'c'	"	'd'
qI	'a', R, 'q2'	'b', R, 'q2'	'c', R, 'q2'	'', R, 'q1'	
<i>q2</i>	'a', R, 'q2'	'b', R, 'q2'	'c', R, 'q2'	' ', <i>L</i> , 'q3'	
<i>q3</i>	'd', R, 'q4'	'd', R, 'q5'	'd', R, 'q6'	'', R, 'q8'	'd', L, 'q3'
<i>q4</i>	'a', R, 'q4'	'b', R, 'q4'	'c', R, 'q4'	'a', L, 'q7'	'd', R, 'q4'
<i>q</i> 5	'a', R, 'q5'	'b', R, 'q5'	'c', R, 'q5'	'b', L, 'q7'	'd', R, 'q5'
<i>q6</i>	'a', R, 'q6'	'b', R, 'q6'	'c', R, 'q6'	'c', L, 'q7'	'd', R, 'q6'
<i>q7</i>	'a', L, 'q7'	'b', L, 'q7'	'c', L, 'q7'		'd', L, 'q3'
<i>q8</i>	'a', N, 'q9'	'b', N, 'q9'	'c', N, 'q9'		' ', R, 'q2'

Описание состояний:

q1 — стартовое состояние для поиска первого символа.

q2 — состояние поиска последнего символа, после нахождения знака ' ' переходит в q3 .

q3 — состояние для чтения текста с конца, после нахождения символа, принадлежащего исходному тексту, меняет его на 'd' и вызывает нужную функцию из q4, q5, q6. Когда встречает символ ' ', переходит в q8.

q4, q5, q6 — состояния, которые ставят справа от исходного текста нужный символ, затем переходят в q7.

q7 — состояние, которое идет по тексту влево, пока не встретит символ 'd', затем возвращается в q3.

q8 – состояние, которое удаляет все символы 'd' из текста.

q9 – конечное состояние.

Перемещения каретки: влево – L, вправо – R, не передвигать – N.

table – словарь таблицы состояний машины Тьюринга, *mem* – начальная строка, *state* – переменная, хранящая текущее состояние, *ind* – переменная, хранящая текущий номер ячейки на ленте.

Разработанный программный код см. в приложении А.

Выводы

Был изучен принцип работы машины Тьюринга. Разработана программа, имитирующая механизм работы машины Тьюринга и решающая поставленную задачу по оборачивании исходной строки.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
L, N, R=-1, 0, 1
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':['d',R,"q4"],'b':['d',R,"q5"],'c':['d',R,"q6"],'d':['d',L,"q3"],'
':[' ',R,"q8"]},
    "q4":
{'a':['a',R,"q4"],'b':['b',R,"q4"],'c':['c',R,"q4"],'d':['d',R,"q4"],'
':['a',L,"q7"]},
    "q5":
{'a':['a',R,"q5"],'b':['b',R,"q5"],'c':['c',R,"q5"],'d':['d',R,"q5"],'
':['b',L,"q7"]},
    "q6":
{'a':['a',R,"q6"],'b':['b',R,"q6"],'c':['c',R,"q6"],'d':['d',R,"q6"],'
':['c',L,"q7"]},
    "q7":
{'a':['a',L,"q7"],'b':['b',L,"q7"],'c':['c',L,"q7"],'d':['d',L,"q3"]},
            {'a':['a',N,"q9"],'b':['b',N,"q9"],'c':['c',N,"q9"],'d':['
    "q8":
',R,"q8"]}
                                           ")
mem = list(" " + input() + "
state = "q1"
ind = 0
while state != "q9":
    symbol = mem[ind]
    next s = table[state][symbol]
   mem[ind] = next s[0]
    ind += next s[1]
    state = next s[2]
print("".join(mem))
```