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. 3344 _____ Мурдасов М.К.

Преподаватель _____ Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Изучение принципа работы машины Тьюринга и конечных автоматов. Применение машины Тьюринга на практике.

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'.**

			a	c	c	a	b	c	b	a	b	a	a	c	a	b			

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:

			b	a	c	a	a	b	a	b	c	b	a	c	c	a			

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.

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

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

Выполнение работы

В первую очередь в переменную *table* был записан словарь, содержащий в себе каждое состояние машины Тьюринга в виде ключей и алгоритмы действий для каждого состояния в виде значений. Сами значения этих ключей – также словари, содержащие в себе алгоритмы действий для каждого возможного символа в ячейке при любом состоянии машины Тьюринга, такие как: символ, записываемый в ячейку, шаг по индексу (влево, вправо, остаться на месте), переход в следующее состояние. Переменная *memory* содержит в себе список, состоящий из символов входной строки (лента). Переменные *q* и *index* содержат в себе начальное состояние и начальный индекс, соответственно.

Немного о состояниях:

q0 – начальное состояние, находит начало строки

q1 – замена символа на «*»

q2 – возвращение к следующему не замененному символу

q3 – запись «a» в начало строки, если замененный символ – «a»

q4 – запись «b» в начало строки, если замененный символ – «b»

q5 – запись «c» в начало строки, если замененный символ – «c»

q6 – удаление всех «*» после переворота строки

q7 – конечное состояние

Далее используется цикл *while*, который, используя данные о текущем состоянии машины Тьюринга, а именно состояния и индекса просматриваемой ячейки, в переменные *symbol*, *delta* и *state* записывает новый символ, шаг по индексу, следующее состояние для машины, соответственно. В состоянии q0 машина Тьюринга доходит до начала строки. В состоянии q1 заменяет первый встречный символ на «*» и переходит в состояние q3, q4 или q5 в зависимости от того, какой символ был заменен и записывает его в начало строки. После чего переходит в состояние q2, чтобы найти следующий для замены символ и опять перейти в состояние q1. Если заменять больше нечего, то из состояния q1 машина переходит в состояние q6. В состоянии q6 она идет от конца строки к началу и стирает все найденные «*», а при нахождении буквы переходит в

состояние q7 и останавливается. Таким образом, получается инвертированная строка, которая выводится программой.

Таблица состояний представлена в табл. 1

Таблица 1 — Таблица состояний

	‘ ’	‘a’	‘b’	‘c’	*
q0	‘ ’; 1; ‘q0’	‘a’; 0; ‘q1’			
q1	‘ ’; -1; ’q6’	‘*’; 0; ’q3’	‘*’; 0; ’q4’	‘*’; 0; ’q5’	‘*’; 1; ’q1’
q2		‘a’; 1; ’q2’	‘b’; 1; ’q2’	‘c’; 1; ’q2’	‘*’; 1; ’q1’
q3	‘a’; 1; ’q2’	‘a’; -1; ’q3’	‘b’; -1; ’q3’	‘c’; -1; ’q3’	‘*’; -1; ’q3’
q4	‘b’; 1; ‘q2’	‘a’; -1; ’q4’	‘b’; -1; ’q4’	‘c’; -1; ’q4’	‘*’; -1; ’q4’
q5	‘c’; 1; ’q2’	‘a’; -1; ’q5’	‘b’; -1; ’q5’	‘c’; -1; ’q5’	‘*’; -1; ’q5’
q6		‘a’; 0; ‘q7’	‘b’; 0; ‘q7’	‘c’; 0; ‘q7’	‘ ’; -1; ‘q6’

Тестирование

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

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

Выводы

Был освоен принцип работы машины Тьюринга. Был написан алгоритм для машины Тьюринга, инвертирующий входную строку.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Murdasov_Mikhail_lb3.py

```
table = {
    'q0':{' ':[' ', 1, 'q0'], 'a':['a', 0, 'q1']},
     'q1':{' ':[' ', -1, 'q6'], 'a':['*', 0, 'q3'], 'b':['*', 0, 'q4'], 'c':['*', 0, 'q5'],'*':['*', 1, 'q1']},
      'q2':{'a':['a', 1, 'q2'], 'b':['b', 1, 'q2'], 'c':['c', 1, 'q2'],'*':['*', 1, 'q1']},
     'q3':{' ':['a', 1, 'q2'], 'a':['a', -1, 'q3'], 'b':['b', -1, 'q3'], 'c':['c', -1, 'q3'],'*':['*', -1, 'q3']},
      'q4':{' ':['b', 1, 'q2'], 'a':['a', -1, 'q4'], 'b':['b', -1, 'q4'], 'c':['c', -1, 'q4'],'*':['*', -1, 'q4']},
      'q5':{' ':['c', 1, 'q2'], 'a':['a', -1, 'q5'], 'b':['b', -1, 'q5'], 'c':['c', -1, 'q5'],'*':['*', -1, 'q5']},
     'q6':{'*':[' ', -1, 'q6'], 'a':['a', 0, 'q7'], 'b':['b', 0, 'q7'], 'c':['c', 0, 'q7']}
}

memory = list(' '*25 + input() + ' '*25)
q = 'q0'
index = 0

while q != 'q7':
    symbol, delta, state = table[q][memory[index]]
    memory[index] = symbol
    index += delta
    q = state

print(''.join(memory))
```