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 _____ Стрижков И.А.

Преподаватель _____ Иванов Д. В.

Санкт-Петербург
2023

Цель работы

Целью работы является реализация машины Тьюринга на Python для моделирования работы вычислительного устройства.

Задание

Вариант 1

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	c	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:

			c	c	c	a	b	c	b	a	b	a	a	c	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.

Ваша программа должна вывести полученную ленту после завершения работ

Выполнение работы

Таблица 1 - Таблица состояний

	'a'	'b'	'c'	' '
q1	'a', R, 'q1_a'	'b', R, 'q2'	'c', R, 'q1_a'	' ', R, 'q1'
q1_a	'a', R, 'q1_a'	'b', R, 'q2'	'c', R, 'q1_a'	' ', L, 'q6'
q2	' ', R, 'q3'	' ', R, 'q3'	' ', R, 'q3'	' ', L, 'q4'
q3	' ', R, 'q5'	' ', R, 'q5'	' ', R, 'q5'	' ', N, 'qT'
q4	'a', R, 'q6'	'b', N, 'q4'	'c', N, 'q4'	' ', N, 'qT'
q5	'a', R, 'q5'	'b', R, 'q5'	'c', R, 'q5'	' ', R, 'qT'
q6	'a', L, 'q6'	'b', L, 'q6'	'c', L, 'q6'	' ', R, 'q4'

Описание состояний:

- q1 - начальное состояние, которое необходимо, чтобы найти первый встретившийся символ ‘a’.
- q2 – состояние, которое определяет символ, следующий за первым встретившимся символом ‘a’.
- q3 – состояние, которое определяет символ после двух символов ‘a’ и отправляет на соответствующий трек состояний.
- q4 - состояние для ‘b’, которое совершает сдвиг влево.
- q5 - состояние для ‘b’, которое совершает сдвиг влево.
- q6 – состояние, которое заменяет символ на ‘b’.
- q7 - состояние для ‘a’, которое совершает сдвиг влево.
- q8 - состояние для ‘a’, которое совершает сдвиг влево.
- q9 – состояние, которое заменяет символ на ‘a’.
- q10 - состояние для ‘c’, которое совершает сдвиг влево.
- q11 - состояние для ‘c’, которое совершает сдвиг влево.
- q12 – состояние, которое заменяет символ на ‘c’.

Принцип работы Машины Тьюринга в коде:

- table – таблица состояний, заданная словарем;
- table_now – текущее состояние, изначально q1;

- С помощью цикла while и таблицы состояний строка изменяется согласно условию.

Разработанный программный код см. в приложении А.

Тестирование

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abcabc	abbc	Программа работает корректно
2.	ccbbaa	ccba	Программа работает корректно
3.	ccacac	cacac	Программа работает корректно

Выводы

Была реализована машины Тьюринга на Python для моделирования работы вычислительного устройства.

С помощью словаря была создана таблица состояний, а с помощью цикла while сымитирована работа машины Тьюринга.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
L, R, N = -1, 1, 0

table = {
     'q1': {'a': ['a', R, 'q1_a'], 'b': ['b', R, 'q2'], 'c': ['c', R, 'q1_a'], ' ': [' ', R, 'q1']},
     'q1_a': {'a': ['a', R, 'q1_a'], 'b': ['b', R, 'q2'], 'c': ['c', R, 'q1_a'], ' ': ['', L, 'q6']},
     'q2': {'a': ['', R, 'q3'], 'b': ['', R, 'q3'], 'c': ['', R, 'q3'], ' ': ['', L, 'q4']},
     'q3': {'a': ['', R, 'q5'], 'b': ['', R, 'q5'], 'c': ['', R, 'q5'], ' ': ['', N, 'qT']},
     'q4': {'a': ['', N, 'qT'], 'b': ['', N, 'qT'], 'c': ['', N, 'qT']},
     'q5': {'a': ['a', R, 'q5'], 'b': ['b', R, 'q5'], 'c': ['c', R, 'q5'], ' ': ['', R, 'qT']},
     'q6': {'a': ['a', L, 'q6'], 'b': ['b', L, 'q6'], 'c': ['c', L, 'q6'], ' ': ['', R, 'q4']},
}

table_now = 'q1'
count = 1
result = list(' ' + input().strip() + ' ')
while table_now != 'qT':
       symbol = result[count]
       result[count] = table[table_now][symbol][0]
       count += table[table_now][symbol][1]
       table_now = table[table_now][symbol][2]
print(''.join(result))
```