**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. 3344 |  | Клюкин А.В. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Изучить работу машины Тьюринга и применить алгоритмы на практике.

## Задание

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.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | “a” | “b” | “c” | “ “ |
| q0 | “a”, N, “q1” | “b”, N, “q1” | “c”, N, “q1” | “ ”, R, “q0” |
| q1 | “ ”, R, “q2” | “ ”, R, “q3” | “ ”, R, “q4” | “ ”, N, “q8” |
| q2 | “a”, R, “q2” | “a”, R, “q3” | “a”, R, “q4” | “ ”, L, “q5” |
| q3 | “b”, R, “q2” | “b”, R, “q3” | “b”, R, “q4” | “ ”, L, “q6” |
| q4 | “c”, R, “q2” | “c”, R, “q3” | “c”, R, “q4” | “ ”, L, “q7” |
| q5 | “a”, L, “q5” | “b”, L, “q5” | “c”, L, “q5” | “a”, R, “q1” |
| q6 | “a”, L, “q6” | “b”, L, “q6” | “c”, L, “q6” | “b”, R, “q1” |
| q7 | “a”, L, “q7” | “b”, L, “q7” | “c”, L, “q7” | “c”, R, “q1” |

Стартовое положение q0. Курсор двигается вправо пока не наткнется на какой-либо символ, кроме пробела. Затем он её перезаписывает, чтобы не было потери и не двигает курсор. Переходит на положение q1 и в зависимости от буквы выбирает алгоритм. Удаляет текущую букву и переходит на следующее положение, записывающее удаленную букву в следующую ячейку. Например, была последовательность abc, буква a удалена и записана в ячейку, в которой находилась b, которая в свою очередь переносится на место элемента c. Когда алгоритм упрётся в конец строки, то происходит шаг назад и в зависимости от положения – q2, q3 или q4, идет переход на позицию q5, q6 или q7. В них алгоритм возвращается в начало до символа пробела и записывает символ, определенный на этапе q2/q3/q4 и возвращается в позицию q1. Так алгоритм уже не удалит первый символ и пройдет строку до конца, увидев пробел при позиции q1 и совершив переход на завершение – q8.

Сама программа состоит из считывания строки, занесенной в словарь таблицы положений и цикл состояний.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
| 1. | abc | cba | Верно |
| 2. | abcabc | cbacba | Верно |
| 3. | abacbbc | cbbcaba | Верно |

## Выводы

Была изучен принцип работы машины Тьюринга и так же был реализован пример использования.

# ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Klyukin\_Aleksandr\_lb4.py

string = list(input())

state = 'q0'

table = {

'q0': {'a': ['a', 0, 'q1'],

'b': ['b', 0, 'q1'],

'c': ['c', 0, 'q1'],

' ': [' ', 1, 'q0']

},

'q1': {'a': [' ', 1, 'q2'],

'b': [' ', 1, 'q3'],

'c': [' ', 1, 'q4'],

' ': [' ', 0, 'q8']

},

'q2': {'a': ['a', 1, 'q2'],

'b': ['a', 1, 'q3'],

'c': ['a', 1, 'q4'],

' ': [' ', -1, 'q5']

},

'q3': {'a': ['b', 1, 'q2'],

'b': ['b', 1, 'q3'],

'c': ['b', 1, 'q4'],

' ': [' ', -1, 'q6']

},

'q4': {'a': ['c', 1, 'q2'],

'b': ['c', 1, 'q3'],

'c': ['c', 1, 'q4'],

' ': [' ', -1, 'q7']

},

'q5': {'a': ['a', -1, 'q5'],

'b': ['b', -1, 'q5'],

'c': ['c', -1, 'q5'],

' ': ['a', 1, 'q1']

},

'q6': {'a': ['a', -1, 'q6'],

'b': ['b', -1, 'q6'],

'c': ['c', -1, 'q6'],

' ': ['b', 1, 'q1']

},

'q7': {'a': ['a', -1, 'q7'],

'b': ['b', -1, 'q7'],

'c': ['c', -1, 'q7'],

' ': ['c', 1, 'q1']

}

}

index = 0

while state != 'q8':

sign, delta, state = table[state][string[index]]

string[index] = sign

index += delta

print(''.join(string))