**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 |  | Сербиновский Ю.М. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы.

Получить навыки составления таблицы машины Тьюринга и работы с машиной Тьюринга.

## 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.

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.

**Ваша программа должна вывести полученную ленту после завершения работы.**

## Выполнение работы.

Сначала была составлена таблица состояний program:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | a | b | c | « » | ! |
| q1 | a;R;q2 | b;R;q2 | c;R;q2 | « »;R;q1 |  |
| q2 | a;R;q2 | a;R;q2 | a;R;q2 | « »;L;q3 |  |
| q3 | « »;R;qa | « »;R;qb | « »;R;qc | « »;R;q! |  |
| qa | a;R;qa | a;R;qa | a;R;qa | « »;L;q4 | !;R;qa |
| qb | a;R;qb | b;R;qb | c;R;qb | « »;L;q4 | !;R;qb |
| qc | a;R;qc | b;R;qc | c;R;qc | « »;L;q4 | !;R;qc |
| q4 | a;L;q4 | b;L;q4 | c;L;q4 | « »;L;q4 | !;L;q4 |
| q! | a;N;q0 | b;N;q0 | c;N;q0 | « »;L;q! | « »;R;q! |

q1 — Начальное состояние, нужно для нахождения начала строки

q2 — Состояние, которое находит конец строки

q3 — Заменяет последний символ строки, который еще не был заменен на « » или «!», на « », в зависимости от замененного символа выберет одно из трех состояний, также машина будет заменять все « » на «!», пока не найдет нужный символ

qa — Находит первый пробел после строки, и заменяет его на «а»

qb — Находит первый пробел после строки, и заменяет его на «b»

qс — Находит первый пробел после строки, и заменяет его на «c»

q4 — Идет влево и ищет следующий пробел, чтобы переключиться на q3 и узнать следующий символ для переноса

q! — Удаляет все «!»

q0 – конечное состояние

Далее следует цикл while, котором считывается symbol, move, state (символ, шаг и следующее состояние), пока машина не перейдет в состояние q0.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| № п/п | Входные данные | Выходные данные |
|  | abcc | ccba |
|  | abbcbacbacc | ccabcabcbba |

## Выводы.

В ходе выполнения лабораторной работы были получены навыки работы с машиной Тьюринга.

# Приложение А Исходный код программы

Название файла: main.py

program = {'q1':{'a':('a', 1, 'q2'), 'b':('b', 1, 'q2'), 'c':('c', 1, 'q2'), ' ':(' ', 1, 'q1')},  
 'q2':{'a':('a', 1, 'q2'), 'b':('b', 1, 'q2'), 'c':('c', 1, 'q2'), ' ':(' ', -1, 'q3')},  
 'q3':{'a':(' ', 1, 'qa'), 'b':(' ', 1, 'qb'), 'c':(' ', 1, 'qc'), ' ':(' ', 1, 'q!')},  
 'qa': {'a': ('a', 1, 'qa'), 'b': ('b', 1, 'qa'), 'c': ('c', 1, 'qa'), ' ': ('a', -1, 'q4'), '!': ('!', 1, 'qa')},  
 'qb': {'a': ('a', 1, 'qb'), 'b': ('b', 1, 'qb'), 'c': ('c', 1, 'qb'), ' ': ('b', -1, 'q4'), '!': ('!', 1, 'qb')},  
 'qc': {'a': ('a', 1, 'qc'), 'b': ('b', 1, 'qc'), 'c': ('c', 1, 'qc'), ' ': ('c', -1, 'q4'), '!': ('!', 1, 'qc')},  
 'q4': {'a': ('a', -1, 'q4'), 'b': ('b', -1, 'q4'), 'c': ('c', -1, 'q4'), ' ': ('!', -1, 'q3'), '!': ('!', -1, 'q4')},  
 'q!': {'a': ('a', 0, 'q0'), 'b': ('b', 0, 'q0'), 'c': ('c', 0, 'q0'), ' ': (' ', 1, 'q!'), '!': (' ', 1, 'q!')}}  
  
  
tape = list(' ' + input() + 13\*' ')  
state = 'q1'  
index = 0  
  
while state != 'q0':  
 symbol, move, state = program[state][tape[index]]  
 tape[index] = symbol  
 index += move  
  
  
  
print(''.join(tape))