**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 конечныне автоматы**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3342 |  | Пушко К.Д. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Изучить принцип работы Машины Тьюринга, написать программу, реализующую работу Машины Тьюринга.

## 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.

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.

## Выполнение работы

Программа получает на вход ленту Машины Тьюринга в виде строки и сохраняет её в виде списка. Алгоритм работы Машины Тьюринга реализован в цикле while, в котором выполняется функционал Машины Тьюринга, посредством обращения к словарю, содержащему таблицу состояний. Состояния с описанием представлены в табл. 1. Цикл продолжает работу, пока не будет достигнуто терминалное состояние завершения выполнения работы Машины Тьюринга. В алфавит были добавлены символы: S, A, B, C, \*. Символ S обозначает начало слова. Символы A, B, C являются временными символами при замене. Они необходимы для того, чтобы программа отличала обработанные и необработанные символы. Символ \* необходим для обозначения уже замененных символов. Результат выводится в строку, и сдвигается, благодаря п.4 из соглашения из условия задачи.

Таблица 1 — состояния Машины Тьюринга

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Состояние | a | b | c | A | B | C | ‘ ’ | \* | S |
| q1 | ['a',L,'q2'] | ['b',L,'q2'] | ['c',L,'q2'] | - | - | - | [' ',R,'q1'] | - | - |
| q2 | - | - | - | - | - | - | ['S',R,'q3'] | - | - |
| q3 | ['a',R,'q3'] | ['b',R,'q3'] | ['c',R,'q3'] | - | - | - | ['\*',N,'q4'], | - | - |
| q4 | ['a',N,'q5'] | ['b',N,'q5'] | ['c',N,'q5'] | ['A',L,'q4'] | ['B',L,'q4'] | ['C',L,'q4'] | - | ['\*',L,'q4'] | ['S',N,'qclear'] |
| q5 | ['\*',N,'qa1'] | ['\*',N,'qb1'] | ['\*',N,'qc1'] | - | - | - | - | - | ['S',L,'qclear'] |
| qa1 | ['a', R, 'qa1'] | ['b', R, 'qa1'] | ['c', R, 'qa1'] | ['A', R, 'qa1'] | ['B', R, 'qa1'] | ['C', R, 'qa1'] | ['A',N,'q4'] | ['\*',R,'qa1'] | - |
| qb1 | ['a', R, 'qb1'] | ['b', R, 'qb1'] | ['c', R, 'qb1'] | ['A', R, 'qb1'] | ['B', R, 'qb1'] | ['C', R, 'qb1'] | ['B',N,'q4'] | ['\*',R,'qb1'] | - |
| qc1 | ['a', R, 'qc1'] | ['b', R, 'qc1'] | ['c', R, 'qc1'] | ['A', R, 'qc1'] | ['B', R, 'qc1'] | ['C', R, 'qc1'] | ['C',N,'q4'] | ['\*',R,'qc1'] | - |
| qclear | - | - | - | ['a',R,'qclear'] | ['b',R,'qclear'] | ['c',R,'qclear'] | [' ',N,'qT'] | [' ',R,'qclear'] | [' ',R,'qclear'] |

q1 – начальное состояние, смещает курсор в первый символ строки.

q2 – устанавливает перед началом строки символ S для обозначения начала строки.

q3 – доходит до конца слова и устанавливает там символ \*, обозначающий конец слова.

q4 – идет до первого встретившегося символа \*.

q5 – определяет какой символ необходимо перенести в конец новой строки.

qa1 – переносит символ a в конец строки.

qb1 – переносит символ b в конец строки.

qc1 – переносит символ c в конец строки.

qclean – удаляет символы S, \*, а так же меняет символы A, B,C на символы a, b, c соответственно.

qT -терминальное состояние, завершающее выполнений цикла.

Разработанный программный код см. в приложении А.

## Тестирование

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| № | Входные данные | Выходные данные |
| 1 | abcabc | cbacba |
| 2 | abacbbc | cbbcaba |
| 3 | acbacbbacbacbaca | acabcabcabbcabca |

## Выводы

Было изучено написание таблицы состояний для машины Тьюринга и написание программы на языке Python, для выполнений операций по Машине Тьюринга.

# Приложение А Исходный код программы

Название файла: main.py

R = 1

N = 0

L = -1

table = {

#дойти до символов

'q1':{' ':[' ',R,'q1'], 'a':['a',L,'q2'], 'b':['b',L,'q2'], 'c':['c',L,'q2'] },

# поставить старт строки

'q2' : {' ':['S',R,'q3']},

# дойти до конца слова

'q3':{' ':['\*',N,'q4'], 'a':['a',R,'q3'], 'b':['b',R,'q3'], 'c':['c',R,'q3'] },

#вернуться до \*

'q4':{'a':['a',N,'q5'], 'b':['b',N,'q5'], 'c':['c',N,'q5'],'\*':['\*',L,'q4'],'S':['S',N,'qclear'],'A':['A',L,'q4'], 'B':['B',L,'q4'], 'C':['C',L,'q4']},

#понять что переносить

'q5':{'S':['S',L,'qclear'], 'a':['\*',N,'qa1'], 'b':['\*',N,'qb1'], 'c':['\*',N,'qc1'],},

#перенос a

'qa1': {'\*':['\*',R,'qa1'],'\*': ['\*', R, 'qa1'], 'a': ['a', R, 'qa1'], 'b': ['b', R, 'qa1'], 'c': ['c', R, 'qa1'],' ':['A',N,'q4'], 'A': ['A', R, 'qa1'], 'B': ['B', R, 'qa1'], 'C': ['C', R, 'qa1']},

#перенос b

'qb1': {'\*':['\*',R,'qb1'],'\*': ['\*', R, 'qb1'], 'a': ['a', R, 'qb1'], 'b': ['b', R, 'qb1'], 'c': ['c', R, 'qb1'],' ':['B',N,'q4'], 'A': ['A', R, 'qb1'], 'B': ['B', R, 'qb1'], 'C': ['C', R, 'qb1']},

#перенос c

'qc1': {'\*':['\*',R,'qc1'],'\*': ['\*', R, 'qc1'], 'a': ['a', R, 'qc1'], 'b': ['b', R, 'qc1'], 'c': ['c', R, 'qc1'],' ':['C',N,'q4'], 'A': ['A', R, 'qc1'], 'B': ['B', R, 'qc1'], 'C': ['C', R, 'qc1']},

'qclear':{'S':[' ',R,'qclear'],'\*':[' ',R,'qclear'],'A':['a',R,'qclear'],'B':['b',R,'qclear'],'C':['c',R,'qclear'],' ':[' ',N,'qT']}

}

lenta = list(' '+input()+' '\*20)

qState = 'q1'

index = 0

while qState != 'qT':

s, step ,st = table[qState][lenta[index]]

lenta[index] = s

index += step

qState=st

print(''.join(lenta).replace(' ',''))