**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 |  | Коняева М.В. |
| Преподаватель |  | Иванов Д.В. |

Санкт-Петербург

2023

## Цель работы

Целью работы является изучение принципа работы машины Тьюринга и конечных автоматов, а также применение машины Тьюринга на практике.

## 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 удаляет в исходной строке два символа, следующих за первым встретившимся символом ‘b’. Если первый встретившийся символ ‘b’ – последний в строке, то удалить его. Если первый встретившийся символ ‘b’ – предпоследний в строке, то удалить один символ, следующий за ним, т. е. последний в строке. Если в строке символ ‘b’ отсутствует, то удалить самый первый символ строки. После удаления в строке не должно оставаться пробелов и пустых мест!

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:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | a | c | c | a | b | a | b | a | a | c | a | b |  |  |  |  |  |

Алфавит: "a", "b", "c", " ".

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.

Программа должна вывести полученную ленту после завершения работы.

## Выполнение работы

В переменную *input* запишем вводимую строку. Переменная *state* отражает номер состояния Машины Тьюринга, *index* - индекс ячейки строки. Был прописан словарь состояний *states*, благодаря которым Машина выполняет поставленную задачу.

Немного о состояниях:

1 – начальное состояние, находит начало строки

2 – нахождение символа «b», если он есть

3 – проверка является «b» последней

4 – проверка является «b» предпоследней

5 – перенос символа в зависимости от стоящей буквы

6 – перенос, если был символ «а»

7 – перенос, если был символ «b»

8 – перенос, если был символ «с»

9 – пропуск символа, для того чтобы понять, какой символ нужно перенести дальше

10 – нахождение начала строки

11 – повторное нахождение «b»

12 – пропуск одного символа назад

13 – перенос символа в зависимости от стоящей буквы

14 – пропуск символа, для того чтобы понять, какой символ нужно перенести дальше

15 – перенос, если был символ «а»

16 – перенос, если был символ «b»

17– перенос, если был символ «с»

18 – нахождение начала строки

19 – удаление символа для состояния 2, 3, 4

Далее используется цикл *while*, в котором считываются данные о текущем состоянии машины Тьюринга, а именно состояния и индекса просматриваемой ячейки, в переменные  *symbol, add, state* записывает новый символ, шаг по индексу, следующее состояние для машины, соответственно. При переходе в состояние 20 машина завершает работу. Таким образом, получается нужная нам строка, которая выводится программой.

Таблица состояний представлена в табл. 1

Таблица 1 — Таблица состояний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ' ' | 'a' | 'b' | 'c' |
| 1 | ' ',1, 1 | 'a', 0, 2 | 'b', 0, 2 | 'c', 0, 2 |
| 2 | ' ',-1, 18 | 'a', 1, 2 | 'b', 1, 3 | 'c', 1, 2 |
| 3 | ' ', -1, 19 | 'a', 1, 4 | 'b', 1, 4 | 'c', 1, 4 |
| 4 | ' ', -1, 19 | 'a', 1, 5 | 'b', 1, 5 | 'c', 1, 5 |
| 5 | ' ', -1, 12 | ' ', -1, 6 | ' ', -1, 7 | ' ', -1, 8 |
| 6 | 'a', 1, 9 | 'a', 1, 9 | 'a', 1, 9 | 'a', 1, 9 |
| 7 | 'b', 1, 9 | 'b', 1, 9 | 'b', 1, 9 | 'b', 1, 9 |
| 8 | 'c', 1, 9 | 'c', 1, 9 | 'c', 1, 9 | 'c', 1, 9 |
| 9 | ' ', 1, 5 | 'a', 1, 5 | 'b', 1, 5 | 'c', 1, 5 |
| 10 | ' ', 1, 11 | 'a', -1, 10 | 'b', -1, 10 | 'c', -1, 10 |
| 11 | ' ', 1, 11 | 'a', 1, 11 | 'b', 1, 14 | 'c', 1, 11 |
| 12 | ' ', -1, 10 |  |  |  |
| 13 | ' ', -1, 20 | ' ', -1, 15 | ' ', -1, 16 | ' ', -1, 17 |
| 14 | ' ', 1, 13 | 'a', 1, 13 | 'b', 1, 13 | 'c', 1, 13 |
| 15 | 'a', 1, 14 | 'a', 1, 14 | 'a', 1, 14 | 'a', 1, 14 |
| 16 | 'b', 1, 14 | 'b', 1, 14 | 'b', 1, 14 | 'b', 1, 14 |
| 17 | 'c', 1, 14 | 'c', 1, 14 | 'c', 1, 14 | 'c', 1, 14 |
| 18 | ' ', 1, 19 | 'a', -1, 18 | 'b', -1, 18 | 'c', -1, 18 |
| 19 |  | ' ', 0, 20 | ' ', 0, 20 | 'c', 0, 20 |

## Тестирование

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | abcabc | abbc | Данные обработаны корректно |
|  | ccbbaa | ccba | Данные обработаны корректно |

## Выводы

В результате проделанной работы был освоен принцип работы и устройство машины Тьюринга. Была написана программа для машины Тьюринга, позволяющая удалять два символа после первого встретившегося символа «b».

# Приложение А Исходный код программы

Название файла: lb3.py

inp = list(input())

state = 1

ind = 0

states = {

1: {" ": [' ', 1, 1], "a": ['a', 0, 2], "b": ['b', 0, 2], "c": ['c', 0, 2]},

2: {" ": [' ', -1, 18], "a": ['a', 1, 2], "b": ['b', 1, 3], "c": ['c', 1, 2]},

3: {" ": [' ', -1, 19], "a": ['a', 1, 4], "b": ['b', 1, 4], "c": ['c', 1, 4]},

4: {" ": [' ', -1, 19], "a": ['a', 1, 5], "b": ['b', 1, 5], "c": ['c', 1, 5]},

5: {" ": [' ', -1, 12], "a": [' ', -1, 6], "b": [' ', -1, 7], "c": [' ', -1, 8]},

6: {" ": ['a', 1, 9], "a": ['a', 1, 9], "b": ['a', 1, 9], "c": ['a', 1, 9]},

7: {" ": ['b', 1, 9], "a": ['b', 1, 9], "b": ['b', 1, 9], "c": ['b', 1, 9]},

8: {" ": ['c', 1, 9], "a": ['c', 1, 9], "b": ['c', 1, 9], "c": ['c', 1, 9]},

9: {" ": [' ', 1, 5], "a": ['a', 1, 5], "b": ['b', 1, 5], "c": ['c', 1, 5]},

10: {" ": [' ', 1, 11], "a": ['a', -1, 10], "b": ['b', -1, 10], "c": ['c', -1, 10]},

11: {" ": [' ', 1, 11], "a": ['a', 1, 11], "b": ['b', 1, 14], "c": ['c', 1, 11]},

12: {" ": [' ', -1, 10]},

13: {" ": [' ', -1, 20], "a": [' ', -1, 15], "b": [' ', -1, 16], "c": [' ', -1, 17]},

14: {" ": [' ', 1, 13], "a": ['a', 1, 13], "b": ['b', 1, 13], "c": ['c', 1, 13]},

15: {" ": ['a', 1, 14], "a": ['a', 1, 14], "b": ['a', 1, 14], "c": ['a', 1, 14]},

16: {" ": ['b', 1, 14], "a": ['b', 1, 14], "b": ['b', 1, 14], "c": ['b', 1, 14]},

17: {" ": ['c', 1, 14], "a": ['c', 1, 14], "b": ['c', 1, 14], "c": ['c', 1, 14]},

18: {" ": [' ', 1, 19], "a": ['a', -1, 18], "b": ['b', -1, 18], "c": ['c', -1, 18]},

19: {"a": [' ', 0, 20], "b": [' ', 0, 20], "c": [' ', 0, 20]}

}

while state != 20:

symbol, add, state = states[state][inp[ind]]

inp[ind] = symbol

ind += add

print(''.join(inp))