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

Студентка гр. 3342 _____ Епонишникова А.И.

Преподаватель _____ Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Целью работы является изучить принцип работы машина Тьюринга и выполнить задание, с помощью полученных знаний.

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}.

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.

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.

Выполнение работы

	a	b	c	‘ ‘
q1	a, R, q2	b, R, q3	c, R, q2	‘ ‘,R, q1
q2	a, R, q2	b, R, q3	c, R, q2	‘ ‘, L, q4
q3	a, R, q7	b, R, q7	c, R, q7	‘ ‘, L, q6
q4	a, L, q4	-	c, L, q4	' ', R, q5
q5	' ', N, qT	-	' ', N, qT	-
q6	-	' ', N, qT	-	-
q7	a, R, q9	b, R, q9	c, R, q9	' ', L, q8
q8	' ', L, qT	' ', L, qT	' ', L, qT	-
q9	a, L, q15	b, L, q16	c, L, q17	' ', L, q18
q10	a, R, q14	a, R, q14	a, R, q14	-
q11	b, R, q14	b, R, q14	b, R, q14	-
q12	c, R, q14	c, R, q14	c, R, q14	-
q13	' ', R, q14	' ', R, q14	' ', R, q14	' ', R, q20
q14	a, R, q19	b, R, q19	c, R, q19	' ', N, q19
q15	a, L, q10	b, L, q10	c, L, q10	' ', L, q10
q16	a, L, q11	b, L, q11	c, L, q11	' ', L, q11
q17	a, L, q12	b, L, q12	c, L, q12	' ', L, q12
q18	a, L, q13	b, L, q13	c, L, q13	' ', L, q13
q19	a, R, q9	b, R, q9	c, R, q9	' ', N, q9
q20	' ', N, qT	' ', N, 'qT	' ', N, qT	' ', N, Qt

R – сдвиг по ленте вправо, L – сдвиг по ленте влево, N – остается на месте

q1 – Поиск по ленте до первой буквы

q2 – Поиск в строчке первой буквы b

q3 – Поиск первой буквы после b

q4 – Возвращение в начало строки, так как символ b не был найден в строке

q5 – Удаление первого символа

q6 – Удаление символа b, так как первый встретившийся символ находился в конце строки

q7 – Поиск второго символа после b

q8 – Удаление последнего символа, так как символ b(который встретился первый раз) был предпоследним в строке

q9 – Поиск символа, который надо переместить на два влево

q10 – Замена символа на a

q11 – Замена символа на b

q12 – Замена символа на c

q13 – Замена символа на ‘ ‘

q14 – шаг вправо для поиска символа для переноса

q15 – шаг влево для замены a

q16 – шаг влево для замены b

q17 – шаг влево для замены c

q18 – шаг влево для замены d

q19 – шаг вправо для поиска символа для переноса

q20 - Перемещение пробела и завершение работы.

qT – конечное состояние

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	abccba	abba
2.	bc	b
3.	acaaccac	caaccac

Выводы

Была изучена работа с машиной Тьюринга, а также реализована программа.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab_3.py

```
     instructions={
          'q1': {' ': [' ', 1, 'q1'], 'a': ['a', 1, 'q2'], 'b': ['b', 1,
'q3'],  'c': ['c', 1, 'q2']},
          'q2': {' ': [' ', -1, 'q4'], 'a': ['a', 1, 'q2'], 'b': ['b', 1,
'q3'], 'c': ['c', 1, 'q2']},
          'q3': {' ': [' ', -1, 'q6'], 'a': ['a', 1, 'q7'], 'b': ['b', 1,
'q7'], 'c': ['c', 1, 'q7']},
          'q4': {' ': [' ', 1, 'q5'], 'a': ['a', -1, 'q4'], 'c': ['c', -
1, 'q4']},
          'q5': {'a': [' ', 0, 'qT'], 'c': [' ', 0, 'qT']},
          'q6': {'b':[' ', 0, 'qT']},
          'q7': {' ': [' ', -1, 'q8'], 'a': ['a', 1, 'q9'], 'b': ['b', 1,
'q9'],  'c': ['c', 1, 'q9']},
          'q8': {'a': [' ', -1, 'qT'], 'b': [' ', -1, 'qT'], 'c': [' ', -
1, 'qT']},
          'q9': {' ': [' ', -1, 'q18'], 'a': ['a', -1, 'q15'], 'b': ['b',
-1, 'q16'], 'c': ['c', -1, 'q17']},
          'q10': {'a': ['a', 1, 'q14'], 'b': ['a', 1, 'q14'], 'c': ['a',
1, 'q14']},
          'q11': {'a': ['b', 1, 'q14'], 'b': ['b', 1, 'q14'], 'c': ['b',
1, 'q14']},
          'q12': {'a': ['c', 1, 'q14'], 'b': ['c', 1, 'q14'], 'c': ['c',
1,  'q14']},
          'q13': {' ': [' ', 1, 'q20'], 'a': [' ', 1, 'q14'], 'b': [' ',
1,  'q14'], 'c': [' ', 1, 'q14']},
          'q14': {' ': [' ', 0, 'q19'], 'a': ['a', 1, 'q19'], 'b': ['b',
1,  'q19'], 'c': ['c', 1, 'q19']},
           'q15': {' ': [' ', -1, 'q10'], 'a': ['a', -1, 'q10'], 'b':
['b', -1, 'q10'], 'c': ['c', -1, 'q10']},
           'q16': {' ': [' ', -1, 'q11'], 'a': ['a', -1, 'q11'], 'b':
['b', -1,  'q11'], 'c': ['c', -1, 'q11']},
           'q17': {' ': [' ', -1, 'q12'], 'a': ['a', -1, 'q12'], 'b':
['b', -1,  'q12'], 'c': ['c', -1, 'q12']},
           'q18': {' ': [' ', -1, 'q13'], 'a': ['a', -1, 'q13'], 'b':
['b', -1,  'q13'], 'c': ['c', -1, 'q13']},
          'q19': {' ': [' ', 0, 'q9'], 'a': ['a', 1, 'q9'], 'b': ['b', 1,
'q9'], 'c': ['c', 1, 'q9']},
           'q20':{'a':[' ', 0, 'qT'],'b':[' ', 0,'qT'],'c':[' ',0,'qT'],'
':[' ', 0,'qT']}

     }

     memory =list(input())
     state = 'q1'
     index = 0
     states=[state]
     while state != 'qT':
         lenta, move, new_state = instructions[state][memory[index]]
         memory[index] = lenta
         index += move
         state = new_state
         states.append(state)
```

```
print(''.join(memory))
```