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

Студентка гр. 2381 _____ Фомина А. К.

Преподаватель _____ Шевская Н.В.

Санкт-Петербург
2022

Цель работы.

Изучить способы работы Машины Тьюринга, написать программу, решающую задачу при помощи неё, а также составить таблицу состояний.

Задание.

Вариант № 2. 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 заменяет в исходной строке символ, идущий после последних двух встретившихся символов 'a', на предшествующий им символ (гарантируется, что это не пробел). Наличие в строке двух подряд идущих символов 'a' гарантируется.

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 15.

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.

Состояние	a	b	c	' ' (пробел)
start	a, R, str	b, R, str	c, R, str	' ', R, start
str	a, R, str	b, R, str	c, R, str	' ', L, found
found_b	a, L, found_ba	b, L, found_b	c, L, found_c	' ', L, end
found_c	a, L, found_ca	b, L, found_b	c, L, found_c	' ', L, end
found_a	a, L, found_aa	b, L, found_b	c, L, found_c	' ', L, end
found_ba	a, L, found_baa	b, L, found_b	c, L, found_c	' ', L, end
found_ca	a, L, found_caa	b, L, found_b	c, L, found_c	' ', L, end
found_aa	a, L, found_aaa	b, L, found_aab	c, L, found_aac	' ', L, end
found_baa	a, L, found_baaa	b, N, end	c, L, found_baac	' ', L, end
found_caa	a, L, found_caaa	b, L, found_caab	c, N, end	' ', L, end
found_aaa	—	—	—	a, N, end
found_baac	a, R, found_baac	c, N, end	—	—
found_baaa	a, R ,found_baaa	a, N, end	—	—
found_caab	a, R, found_caab	—	b, N, end	—
found_caaa	a, R, found_caaa	—	a, N, end	—
found_aab	a, R, 'found_aab'	—	—	b, N, end
found_aac	a, R, found_aac	—	—	c, N, end

start – начало, ищется начальная буква

str – прохождение по всей строке до конца, пока не встретиться символ пробела

found_b – была найдена буква b, ищется буква, идущая до неё

found_c – была найдена буква c, ищется буква, идущая до неё

found_a – была найдена буква a, ищется буква, идущая до неё

found_ba – было найдено буквосочетание ba, ищется предыдущая буква a

found_ca – было найдено буквосочетание ca, предыдущая буква a

found_aa – было найдено буквосочетание aa, ищется предыдущая буква a

found_baa – было найдено буквосочетание baa, ищется буква, идущая до

found_caa – было найдено буквосочетание caa, ищется буква, идущая до, если это “c”, то переход в конечное состояние

found_aaa – было найдено буквосочетание aaa, замена символа пробела на “a”

found_baac – было найдено буквосочетание baac, замена “b” на “c”

found_baab – было найдено буквосочетание baab, замена “b” на “b”

found_baaa – было найдено буквосочетание baaa, замена “b” на “a”

found_caab – было найдено буквосочетание caab, замена “c” на “b”

found_caaa – было найдено буквосочетание caaa, замена “c” на “a”

found_aab – было найдено буквосочетание aab, замена символа пробела на “b”

found_aac – было найдено буквосочетание aac, замена символа пробела на “c”

end – конечное состояние

Объявляются переменные *L*, *R*, *N* для перемещения машины. В *memory* записывается изначальная строка. В *state* записывается начальное состояние “start”. В ind записывается начальная позиция на ленте. В цикле, пока значение не станет равно конечному состоянию, записывается символ, состояние и шаг. Обновляются символ и состояние, делается шаг. После выполнения цикла, программа выводит измененную строку на экран.

Разработанный программный код см. в приложении А.

Результаты тестирования см. в приложении Б.

Вывод.

Были изучены способы работы Машины Тьюринга. Полученные знания были применены на практике для составления таблицы состояний и решения задачи.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main_lb3.py

```
L, R, N = -1, 1, 0

table = {'start': {'a': ['a', R, 'str'], 'b': ['b', R, 'str'], 'c': ['c', R, 'str'],
                   ' ': [' ', R, 'start']},
         'str': {'a': ['a', R, 'str'], 'b': ['b', R, 'str'], 'c': ['c', R, 'str'], ' ': [' ', L, 'found']},
         'found': {'a': ['a', L, 'found_a'], 'b': ['b', L, 'found_b'], 'c': ['c', L, 'found_c'], ' ': [' ', N, 'end']},
         #случай замены пробела на символ
         'found_a': {'a': ['a', L, 'found_aa'], 'b': ['b', L, 'found_b'], 'c': ['c', L, 'found_c'], ' ': [' ', N, 'end']},
         'found_aa': {'a': ['a', R, 'found_aaa'], 'b': ['b', R, 'found_aab'], 'c': ['c', R, 'found_aac'], ' ': [' ', N, 'end']},
         'found_aaa': {' ': ['a', N, 'end'], 'a': ['a', R, 'found_aaa']},
         'found_aab': {' ': ['b', N, 'end'], 'a': ['a', R, 'found_aab']},
         'found_aac': {' ': ['c', N, 'end'], 'a': ['a', R, 'found_aac']},
         #случай замены b
         'found_b': {'a': ['a', L, 'found_ba'], 'b': ['b', L, 'found_b'], 'c': ['c', L, 'found_c'], ' ': [' ', N, 'end']},
         'found_ba': {'a': ['a', L, 'found_baa'], 'b': ['b', L, 'found_b'], 'c': ['c', L, 'found_c'], ' ': [' ', N, 'end']},
         'found_baa': {'b': ['b', N, 'end'], 'c': ['c', R, 'found_baac'], 'a': ['a', R, 'found_baaa'], ' ': [' ', N, 'end']},
         'found_baac': {'b': ['c', N, 'end'], 'a': ['a', R, 'found_baac']},
         'found_baaa': {'b': ['a', N, 'end'], 'a': ['a', R, 'found_baaa']},
         #случайн замены c
         'found_c': {'a': ['a', L, 'found_ca'], 'b': ['b', L, 'found_b'], 'c': ['c', L, 'found_c'], ' ': [' ', N, 'end']},
         'found_ca': {'a': ['a', L, 'found_caa'], 'b': ['b', L, 'found_b'], 'c': ['c', L, 'found_c'], ' ': [' ', N, 'end']},
```

```
            'found_caa': {'a': ['a', R, 'found_caaa'], 'b': ['b', R,
'found_caab'], 'c': ['c', N, 'end'], ' ': [' ', N, 'end']},
            'found_caab': {'c': ['b', N, 'end'], 'a': ['a', R,
'found_caab']},
            'found_caaa': {'c': ['a', N, 'end'], 'a': ['a', R,
'found_caaa']},
            }
    memory = list(input())

    state = 'start'
    ind = 0
    states = [state]
    while state != 'end':
        symbol, i, state = table[state][memory[ind]]
        memory[ind] = symbol
        ind +=i
        states.append(state)

    print(*memory, sep='')
```

ПРИЛОЖЕНИ Б
ТЕСТИРОВАНИЕ

Таблица Б.1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abcaabc	abcaacc	Верный ответ
2.	bcaabaac	bcaabaab	Верный ответ
3.	aabbaa	aabbaab	Верный ответ