

**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. 3341

Чинаева М. Р.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

## **Цель работы**

Написать программу, которая заменяет в исходной строке символ, идущий после последних двух встретившихся символов 'a', на предшествующий им символ, используя машину Тьюринга.

## 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.

Для примера выше лента будет выглядеть так:

acaabcbabaabab

Алфавит: abc" " (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.

## **Основные теоретические положения**

Машина Тьюринга (МТ) состоит из двух частей: неподвижной бесконечной ленты (памяти) и автомата (процессора).

1. Лента используется для хранения информации. Она бесконечна в обе стороны и разбита на клетки, которые никак не нумеруются и не именуется. В каждой клетке может быть записан один символ или ничего не записано.

2. Алфавит ленты - конечное множество всех возможных символов ленты.

3. Автомат – это активная часть Машины Тьюринга. В каждый момент он размещается под одной из клеток ленты и видит её содержимое; это видимая клетка, а находящийся в ней символ – видимый символ; содержимое же соседних и других клеток автомат не видит. Кроме того, в каждый момент автомат находится в одном из состояний. Существует конечное число таких состояний.

В каждом из состояний автомат выполняет какую-то конкретную операцию. Существует заключительное состояние, в котором автомат останавливается.

Автомат за один такт (шаг) может выполнить следующие действия :

1. считать видимый символ;
2. записывать в видимую клетку новый символ (в том числе пустой символ);
3. сдвигаться на одну клетку влево или вправо («перепрыгивать» сразу через несколько клеток автомат не может);
4. перейти в следующее состояние.

## Выполнение работы

Для выполнения поставленной задачи была написана таблица состояний (см. Таблица 1). В каждой её ячейке написаны через запятую:

1. Символ, на который заменят текущий символ;
2. Направление, куда нужно сдвинуться: R- вправо, N – не двигаться, L – влево;
3. На какое состояние следует переместиться.

Таблица 1 – Таблица состояний машины Тьюринга

| Состояние | “a”          | “b”          | “c”          | “ ”          |
|-----------|--------------|--------------|--------------|--------------|
| “q1”      | “a”, R, “q2” | “b”, R, “q2” | “c”, R, “q2” | “ ”, R, “q1” |
| “q2”      | “a”, R, “q2” | “b”, R, “q2” | “c”, R, “q2” | “ ”, R, “q3” |
| “q3”      | “a”, L, “q4” | “b”, L, “q3” | “c”, L, “q3” |              |
| “q4”      | “a”, L, “q5” | “b”, L, “q3” | “c”, L, “q3” |              |
| “q5”      | “a”, R, “q6” | “b”, R, “q7” | “c”, R, “q8” |              |
| “q6”      | “a”, R, “q6” | “a”, N, “q9” | “a”, N, “q9” | “a”, N, “q9” |
| “q7”      | “a”, R, “q7” | “b”, N, “q9” | “b”, N, “q9” | “b”, N, “q9” |
| “q8”      | “a”, R, “q8” | “c”, N, “q9” | “c”, N, “q9” | “c”, N, “q9” |

Опишем подробнее назначение каждого состояния:

1. q1 - Начальное состояние, чтобы найти начало строки. Переход на состояние q2 при обнаружении символа 'a', 'b', 'c', либо сразу смещение вправо, если встречен пробел.

2. q2 – Поиск конца строки. В случае обнаружения пробела, переход на состояние q3.

3. q3 - Поиск ближайшего (считая слева) символа 'a'. После нахождения этого символа, смещение влево на один символ и переход к состоянию q4.

4. q4 - Проверка символа после символа 'a'. Если этот символ 'a', значит найдены последние встречающиеся в строке подряд символы 'a', совершается переход к состоянию q5, если нет, смещение влево на один символ и возврат к состоянию q3.

5. q5 - "Запоминание" символа перед последними двумя символами 'a' и переход к соответствующему выполняющему операции состоянию q6, q7 или q8.

6. q6 - Заменяет следующий символ после последних двух символов 'a' на 'a'.

7. q7 - Заменяет следующий символ после последних двух символов 'a' на 'b'.

8. q8 - Заменяет следующий символ после последних двух символов 'a' на 'c'.

Код осуществляет обработку каждого состояния и переходов между ними, в результате изменяя входную ленту согласно условиям задачи.

Функция:

```
def turing_machine(tape, current_state, index)
```

Получает на вход ленту(список введенных символов), текущее состояние машины, индекс ячейки, на которой находится каретка.

С помощью цикла while машина Тьюринга меняет состояния, пока не дойдет до конечного. Сначала в список now считываются данные, о действиях, которые надо совершить машине. Значение на ленте меняется на заданное состоянием. Если состояние конечное производится выход из цикла. Если же этого не произошло индекс сдвигается на значение, заданное now[MOVE\_TO], текущее состояние меняется на now[NEXT\_Q].

Функция возвращает список символов на ленте.

Работа программы:

Инициализация переменных R, L, N, обозначающих изменение индекса ячейки, на которой находится каретка, CHANGE\_TO, MOVE\_TO, NEXT\_Q, обозначающих индекс элемента, нужного для действия.

Считывается лента из потока ввода и приводится к типу list.

Задается начальное состояние current\_state='q1'

Инициализируется значение индекса ячейки, на которой находится каретка, равное 0.

На экран выводится результат функции `turing_machine(tape, current_state, index)`, преобразованный к строке.

## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

| № п/п | Входные данные | Выходные данные |
|-------|----------------|-----------------|
| 1.    | abcaabc        | abcaacc         |
| 2.    | abaa           | abaab           |



## **Выводы**

В результате работы было продемонстрировано применение Машины Тьюринга для решения задачи замены символа в строке.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

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

CHANGE_TO = 0
MOVE_TO = 1
NEXT_Q = 2

program = {
    'q1' : {'a' : ['a', R, 'q2'], 'b' : ['b', R, 'q2'], 'c' : ['c',
R, 'q2'], ' ' : [' ', R, 'q1']},
    'q2' : {'a' : ['a', R, 'q2'], 'b' : ['b', R, 'q2'], 'c' : ['c',
R, 'q2'], ' ' : [' ', L, 'q3']},
    'q3' : {'a' : ['a', L, 'q4'], 'b' : ['b', L, 'q3'], 'c' : ['c',
L, 'q3']},
    'q4' : {'a' : ['a', L, 'q5'], 'b' : ['b', L, 'q3'], 'c' : ['c',
L, 'q3']},
    'q5' : {'a' : ['a', R, 'q6'], 'b' : ['b', R, 'q7'], 'c' : ['c',
R, 'q8']},
    'q6' : {'a' : ['a', R, 'q6'], 'b' : ['a', N, 'q9'], 'c' : ['a',
N, 'q9'], ' ' : ['a', N, 'q9']},
    'q7' : {'a' : ['a', R, 'q7'], 'b' : ['b', N, 'q9'], 'c' : ['b',
N, 'q9'], ' ' : ['b', N, 'q9']},
    'q8' : {'a' : ['a', R, 'q8'], 'b' : ['c', N, 'q9'], 'c' : ['c',
N, 'q9'], ' ' : ['c', N, 'q9']}
}

def turing_machine(tape, current_state, index):
    while True:
        now=program[current_state][tape[index]]
        tape[index]=now[CHANGE_TO]
        if now[NEXT_Q]=='q9':
            break
        index+= now[MOVE_TO]
        current_state = now[NEXT_Q]
    return tape

tape=list(input())
current_state='q1'
index=0
print(''.join(turing_machine(tape, current_state, index)))
```