

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Информатика»**  
**Тема: Машина Тьюринга**

Студентка гр. 2381

\_\_\_\_\_

Слабнова Д.А.

Преподаватель

\_\_\_\_\_

Шевская Н.В.

Санкт-Петербург

2022

### **Цель работы.**

Смоделировать работу машины Тьюринга и написать программу, которая заменяет в исходной строке символ, предшествующий первому встретившемуся символу 'с' на символ, следующий за первым встретившимся символом 'а'. Если первый встретившийся символ 'а' в конце строки, то использовать в качестве заменяющего.

### **Основные теоретические положения.**

Машина Тьюринга — машина, состоящая из двух частей: ленты — памяти, обрабатываемой автоматом, и разделённой на ячейки; и самого автомата — устройства, которое обрабатывает информацию на ленте и может совершать только три действия: записать символ в ячейку ленты, перейти в клетку справа/слева от текущей или остаться на месте, перейти в другое состояние.

Алфавит ленты — символы, которые могут быть в ленте.

Состояние автомата определяет то, какие операции с какими символами он будет проделывать. Часто работу автомата (алгоритм по которому действует автомат), можно представить в виде таблицы, где столбцы отвечают каждый за свой символ алфавита, а строки — за состояния. Так, «посмотрев» на ячейку можно понять, что будет делать автомат в конкретном состоянии с конкретной буквой.

### **Ход работы.**

Алфавит ленты в данной работе состоит из символов «а», «b», «с» и « ». Программа должна заменить символ слева от первого символа «с» на символ справа от первого символа «а». На основе этого была составлена таблица состояний №1.

Таблица 1: Таблица состояний №1

	«a»	«b»	«c»	« »
q0	«a», R, q1	«c», R, q0	«c», R, q0	« », R, q0
q1	«a», L, qa-	«b», L, qb-	«c», L, qc-	« », L, qa-
qa-	«a», L, qa-	«b», L, qa-	«c», L, qa-	« », R, qa+
qa+	«a», R, qa+	«b», R, qa+	«c», L, qaw	
qaw	«a», N, qend	«a», N, qend	«a», N, qend	«a», N, qend
qb-	«a», L, qb-	«b», L, qb-	«c», L, qb-	« », R, qb+
qb+	«a», R, qb+	«b», R, qb+	«c», L, qbw	
qbw	«b», N, qend	«b», N, qend	«b», N, qend	«b», N, qend
qc-	«a», L, qc-	«b», L, qc-	«c», L, qc-	« », R, qc+
qc+	«a», R, qc+	«b», R, qc+	«c», L, qcw	
qcw	«c», N, qend	«c», N, qend	«c», N, qend	«c», N, qend

«q0» — начальное состояние: «Идём вправо до тех пор, пока не найдём a, после чего состояние меняется на q1»

«q1» — состояние в котором мы «запоминаем» символ справа от первой a (переходим в разные состояния в зависимости от «запоминаемого символа»: qa-, qb- или qc-) и переходим в ячейку слева.

«qa-», «qb-», «qc-» - идём влево до начала строки, и когда «доходим» меняем состояние на «qa+», «qb+» или «qc+» соответственно и переходим на ячейку справа.

«qa+», «qb+», «qc+» - идём вправо до первого символа «c», после чего меняем состояние на «qaw», «qbw» или «qcw» соответственно и переходим в ячейку слева.

«qaw», «qbw», «qcw» - записываем в текущую ячейку символ «a», «b» или «c» соответственно и переходим в состояние «qend».

«qend» - конечное состояние, автомат заканчивает свою работу и выводит результат.

### Выводы.

Из-за невозможности автомата «запоминать» значения в ячейках и перемещаться по указанному адресу, работать с ним «неудобно». В данной

работе, чтобы обеспечить «запоминание» символа после первого символа «а» пришлось создать три набора по своей сути одинаковых состояний ( $q_i^-$ ,  $q_i^+$ ,  $q_i^w$ ).

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

НАЗВАНИЕ ФАЙЛА: CS\_LAB\_3.PY

```
lenta = list(input())
tur = { 'q0': {'a': ['a', 1, 'q1'], 'b': ['b', 1, 'q0'], 'c': ['c', 1,
'q0'], ' ': [' ', 1, 'q0']}},
      'q1': {'a': ['a', -1, 'qa-'], 'b': ['b', -1, 'qb-'], 'c': ['c', -1,
'qc-'], ' ': [' ', -1, 'qa-']}},
      'qa-': {'a': ['a', -1, 'qa-'], 'b': ['b', -1, 'qa-'], 'c': ['c', -1,
'qa-'], ' ': [' ', 1, 'qa+']}},
      'qa+': {'a': ['a', 1, 'qa+'], 'b': ['b', 1, 'qa+'], 'c': ['c', -1,
'qaw'], ' ': [' ', 0, 'qend']}},
      'qaw': {'a': ['a', 0, 'qend'], 'b': ['a', 0, 'qend'], 'c': ['a', 0,
'qend'], ' ': ['a', 0, 'qend']}},
      'qb-': {'a': ['a', -1, 'qb-'], 'b': ['b', -1, 'qb-'], 'c': ['c', -1,
'qb-'], ' ': [' ', 1, 'qb+']}},
      'qb+': {'a': ['a', 1, 'qb+'], 'b': ['b', 1, 'qb+'], 'c': ['c', -1,
'qbw'], ' ': [' ', 0, 'qend']}},
      'qbw': {'a': ['b', 0, 'qend'], 'b': ['b', 0, 'qend'], 'c': ['b', 0,
'qend'], ' ': ['b', 0, 'qend']}},
      'qc-': {'a': ['a', -1, 'qc-'], 'b': ['b', -1, 'qc-'], 'c': ['c', -1,
'qc-'], ' ': [' ', 1, 'qc+']}},
      'qc+': {'a': ['a', 1, 'qc+'], 'b': ['b', 1, 'qc+'], 'c': ['c', -1,
'qcw'], ' ': [' ', 0, 'qend']}},
      'qcw': {'a': ['c', 0, 'qend'], 'b': ['c', 0, 'qend'], 'c': ['c', 0,
'qend'], ' ': ['c', 0, 'qend']}
    }
sost = 'q0'
i=0
move = 0
write = ''
cond_mass = ['q0']
while(sost != 'qend'):
    symbol, direction, sost = tur[sost][lenta[i]]
    lenta[i] = symbol
    i += direction
    cond_mass.append(sost)
print("".join(lenta))
```