

**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

Антипина В.А.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

## **Цель работы**

Изучить принцип работы машины Тьюринга и научиться создавать для неё алгоритмы.

## 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 длина 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.

Ваша программа должна вывести полученную ленту после завершения работы.

В отчете предоставьте таблицу состояний. Отдельно кратко опишите каждое состояние, например:

q1 - начальное состояние, которое необходимо, чтобы найти первый встретившийся символ 'b'.

## Основные теоретические положения

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

Лента используется для хранения информации. Она бесконечна в обе стороны и разбита на клетки, которые никак не нумеруются и не именуются. В каждой клетке может быть записан один символ или ничего не записано. Память пассивна: она ничего не делает, просто хранит данные.

Алфавит ленты - конечное множество всех возможных символов ленты. Если предположить, что видимые символы - весь алфавит ленты из примера выше, то мы имеем следующий алфавит:  $\{1, 0, +, 'a', '\ '\}$ . Последний символ - пустой, означает пустое содержимое клетки.

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, которые обычно обозначаются буквой  $q$  с номерами:  $q_0, q_1, q_2$  и т.д. 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

Был создан словарь, ключами которого являлись состояния машины Тьюринга, а значениями — словари, ключи которых, в свою очередь, — символы алфавита, а значения — кортежи из «команд» для машины.  $Q_0$  — начальное состояние, которое необходимо для того, чтобы найти первый встретившийся символ „b“. В этом состоянии значения не перезаписываются, а каретка двигается вправо, пока не найдёт искомый символ. В этом случае она также переходит в правую ячейку, а состояние машины меняется на  $q_1$ . Состояние  $q_1$  заменяет следующий за „b“ символ на пробел («пустую» ячейку) и переводит машину в состояние  $q_2$ , если за „b“ не стоял пробел (то есть если „b“ не был последним символом в строке). В этом случае машина переходит в состояние  $q_3$ , а каретка двигается на ячейку влево. В состоянии  $q_2$  заменяется второй встретившийся за „b“ символ на пробел и прекращается работа аппарата (осуществляется переход в терминальное состояние  $q_4$ ). В состоянии  $q_3$  „b“ заменяется на пробел, после чего машина прекращает работу.

Входные данные были записаны в список `memory`. `Index` присвоено значение 0, а состояние было установлено  $q_0$ . В цикле `while` (который работает, пока машина не будет в терминальном состоянии или пока строка не закончится (последнее условие может завершить цикл, если „b“ в строке нет)). `Symbol` — это символ полученной на вход строки, переменные `new_symbol`, `delta`, `state` соответствуют символу, на который нужно заменить символ в ячейке, перемещению каретки (1 — вправо на ячейку, -1 — влево, 0 - оставить в текущей ячейке). Значение ячейки перезаписывается в списке `memory`, `index` увеличивается на `delta`.

Если по завершении работы цикла состояние осталось начальным, „b“ в строке не встретилось. Тогда удаляется первый элемент списка, не являющийся пробелом. Программа выводит изменённый список без разделителей.

Разработанный программный код см. в приложении А.



## Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	abc	ab	Программа работает корректно
2.	aaaaaас	aaaaас	
3.	abc	ab	



## **Выводы**

Был изучен принцип работы машины Тьюринга. Была написана программа, имитирующая работу машины Тьюринга, которая получает на вход строку и преобразовывает её по заданному алгоритму. В работе были использованы словари, списки и цикл `while`.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Antipina\_Veronika\_lb3.py

```
table = {
    'q0':{
        'a':('a',1,'q0'),
        'b':('b',1,'q1'),
        'c':('c',1,'q0'),
        ' ':(' ',1,'q0'),
    },
    'q1':{
        'a':('',1,'q2'),
        'b':('',1,'q2'),
        'c':('',1,'q2'),
        ' ':('',-1,'q3'),
    },
    'q2':{
        'a':('',0,'q4'),
        'b':('',0,'q4'),
        'c':('',0,'q4'),
        ' ':('',0,'q4'),
    },
    'q3':{
        'b':('',0,'q4'),
    },
}

memory = list(input())
index = 0
state = 'q0'

while(state!='q4' and index<len(memory)):
    symbol = memory[index]
    new_symbol,delta,state = table[state][symbol]
    memory[index] = new_symbol
    index+=delta

if(state == 'q0'):
    for i in range(0,len(memory)):
        if(memory[i]!=' '):
            memory[i] = ' '
            break

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