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

Самокрутов А.Р.
Иванов Д.В.

Санкт-Петербург

2023

Цель работы

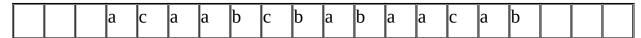
Целью лабораторной работы является изучение работы конечных автоматов, в частности — машины Тьюринга. Для этого требуется написать программу на языке программирования Python, которая моделирует работу машины Тьюринга и выполняет определённую задачу.

Задание

Вариант 2

На вход программе подается строка неизвестной длины. Каждый элемент является значением в ячейке памяти ленты Машины Тьюринга.

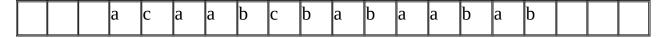
На ленте находится последовательность латинских букв из алфавита {a, b, c}.



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:



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.

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

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

Выполнение работы

Объявляются константы R, L и N соответственно равные 1, -1 и 0. В данной работе лента машины Тьюринга будет рассматриваться как список, образованный элементами строки, считанной программой из потока ввода. Движению вправо по ленте будет соответствовать инкрементация индекса элемента списка, движению влево — декрементация, остановке на месте — отсутствие изменения индекса. Далее инициализируются константы $CHANGE_TO$, $MOVE_TO$ и $NEXT_STATE$ со значениями 0, 1 и 2. Их смысл будет раскрыт далее в тексте.

Создаётся функция turing_machine() с аргументами program (словарь, ключи которого являются состояниями программы, а значения — словари типа элемент ленты — кортеж из трёх элементов: первый элемент — какой символ необходимо записать в текущую ячейку, второй — в каком направлении двигаться, третий — в какое состояние перейти. Индексу каждого из этих элементов-действий соответствует переменная *CHANGE_TO*, *MOVE_TO* и *NEXT_STATE*), tape (массив символов, полученный из строки, поданной в поток ввода), start_state (начальное состояние машины), current_cell (номер клетки, в которой размещён автомат), end_state (конечное состояние).

Функция turing_machine(program, tape, start_state, current_cell, end_state).

Значение start_state — начального состояния — записывается в переменную current_state — текущее состояние. В бесконечном цикле объявляется переменная rows со значением, полученным из словаря program по ключу current_state — словарь типа символ алфавита — кортеж из элементовдействий автомата; объявляется переменная current_row, ей присваивается значение словаря rows по ключу tape[current_cell] (символ в ленте, на который в данный момент указывает автомат); значение текущей ячейки ленты изменяется на current_row[CHANGE_TO]; если в current_row[MOVE_TO] находится 0, т.е. N, то цикл прерывается — программа останавливается; к значению current_cell прибавляется значение current_row[MOVE_TO] — 1 или -1, т.е. R или L; переменной current state присваивается значение current row[NEXT STATE].

Функция возвращает строку, полученную из списка *tape* методом *join()*. Далее создаётся переменная *program*, в которой описывается программа машины Тьюринга. Ей соответствует следующая таблица:

	'a'	'b'	'C'	11
q_0	'a'; R; q ₁	'b'; R; q ₁	'c'; R; q ₁	' '; R; q ₀
q_1	'a'; R; q ₁	'b'; R; q ₁	'c'; R; q ₁	''; L; q ₂
q_2	'a'; L; q ₃	'b'; L; q ₂	'c'; L; q ₂	
q_3	'a'; L; q ₄	'b'; L; q ₂	'c'; L; q ₂	
q ₄	'a'; R; q ₅	'b'; R; q ₈	'c'; R; q ₁₁	
q ₅	'a'; R; q ₆			
q_6	'a'; R; q ₇			
q ₇	'a'; N; q ₁₄			
q_8	'a'; R; q ₉			
q ₉	'a'; R; q ₁₀			
q ₁₀	'b'; N; q ₁₄			
q ₁₁	'a'; R; q ₁₂			
q ₁₂	'a'; R; q ₁₃			
q ₁₃	'c'; N; q ₁₄			

Описание каждого состояния:

 q_0 ('start') — начальное состояние. Автомат движется вправо пока не встретит символ, отличный от пробела (пустой клетки). При встрече такого символа автомат движется вправо и переходит в состояние q_1 .

 q_1 ('skip through') — автомат движется по вправо по строке пока не дойдёт до её конца. Считав символ пробела, автомат сдвигается на клетку левее и переходит в состояние q_2 .

 q_2 ('step back') — автомат движется влево по строке пока не встретит символ 'a'. В этом случае он сдвигается влево и переходит в состояние q_3 .

q₃ ('found first a') — автомат уже нашёл символ 'a' в предыдущей ячейке. Если в текущей клетке символ 'a', то машина переходит в состояние q₄, иначе — в q₂, и сдвигается влево.

 q_4 ('found second a') — автомат уже нашёл второй символ 'a' в предыдущей ячейке. Если автомат считал символ 'a', 'b' или 'c', то он переходит в состояние q_5 , q_8 , или q_{11} соответственно, двигаясь вправо.

 q_5 ('read an a') — на предыдущем шаге автомат прочитал символ 'a' и перемещается в направлении к символу справа от 'aa'. Автомат перемещается вправо и переходит в состояние q_6 .

 q_6 ('read an a, skip') — автомат передвигается через второй из двух символов 'a'. Он перемещается вправо и переходит в состояние q_7 .

 q_7 ('replace with a') — автомат заменяет содержимое текущей ячейки на 'a' и останавливается, переходя в состояние q_{14} .

 q_8 ('read a b') — на предыдущем шаге автомат прочитал символ 'b' и перемещается в направлении к символу справа от 'aa'. Автомат перемещается вправо и переходит в состояние q_9 .

 q_9 ('read a b, skip') — автомат передвигается через второй из двух символов 'a'. Он перемещается вправо и переходит в состояние q_{10} .

 q_{10} ('replace with b') — автомат заменяет содержимое текущей ячейки на 'b' и останавливается, переходя в состояние q_{14} .

 q_{11} ('read a c') — на предыдущем шаге автомат прочитал символ 'c' и перемещается в направлении к символу справа от 'aa'. Автомат перемещается вправо и переходит в состояние q_{12} .

 q_{12} ('read a c, skip') — автомат передвигается через второй из двух символов 'a'. Он перемещается вправо и переходит в состояние q_{10} .

 q_{13} ('replace with c') — автомат заменяет содержимое текущей ячейки на 'b' и останавливается, переходя в состояние q_{14} .

 $q_{14}('end')$ — конечное состояние автомата.

Разработанный программный код см. в приложении А.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

No	Входные данные	Выходные данные	Комментарии
п/п			
1.	abcabcaabc	abcabcaacc	
2.	baa	baa	Проверка случая, когда символ, который нужно заменить, находится вне исходной строки.
3.	aacaacaab	Aacaacaac	Проверка корректной работы программы со строкой, в которой 'аа' встречается более одного раза.

Выводы

Была разработана программа на языке программирования Python, симулирующая работу машины Тьюринга. Была описана программа машины Тьюринга, с помощью которой автомат способен обработать строку символов определённым образом.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
Название файла: lab.py
    R = +1
    L = -1
    N = 0
    CHANGE_TO = 0
    MOVE_TO = 1
    NEXT_STATE = 2
    def turing_machine(program, tape, start_state, current_cell,
end_state):
       current_state = start_state
       while True:
          rows = program[current_state]
          current_row = rows[tape[current_cell]]
          tape[current_cell] = current_row[CHANGE_T0]
          if not current_row[MOVE_TO]:
             break
          current_cell += current_row[MOVE_T0]
          current_state = current_row[NEXT_STATE]
       return ''.join(tape)
    program = {'start': {'a': ('a', R, 'skip through'), 'b': ('b', R,
('b', R, 'skip through'), 'c': ('c', R, 'skip through'), ' ': (' ', L,
'step back')},
             'step back': {'a': ('a', L, 'found first a'), 'b': ('b',
L, 'step back'),
             'c': ('c', L, 'step back')},
             'found first a': {'a': ('a', L, 'found second a'), 'b':
('b', R, 'read a b'), 'c': ('c', R, 'read a c')},
             'read an a': {'a': ('a', R, 'read an a, skip')},
             'read an a, skip': {'a': ('a', R, 'replace with a')},
  'replace with a': {'a': ('a', N, 'end'), 'b': ('a', N,
tape = list(str(input()) + ' ')
    print(turing_machine(program, tape, 'start', 0, 'end')[:-1])
```