

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

Студент гр. 3344

Щербак М.С.

Преподаватель

Иванов Д.В.

Санкт-Петербург

2023

Цель работы

Изучение и реализация базовых принципов работы машины Тьюринга на языке Python.

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 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

Машина переходит в состояние `qend`. Основное тело программы: считывается строка и приводится к списку. Задается начальное состояние и позиция курсора. Выполняется цикл пока состояние не будет `qend`. В переменные считываются соответствующее значение. На изначальное место символа, ставится символ указанный в словаре. Потом машина переходит в другое состояние или же остается в текущем. Выводится полученная лента. Получает символ из текущей ячейки ленты и находит соответствующее действие в словаре `states` для текущего состояния. Символ в ячейке ленты заменяется на новый символ, указанный в словаре `states` для текущего состояния. В начале определяются значения переменных `R`, `L`, `N` и создается словарь `states`, который содержит описание состояний машины Тьюринга и действий, которые нужно выполнить при переходе из одного состояния в другое. Затем программа запрашивает у пользователя строку. `Q0` работает пока мы не найдем первый символ, `q1` пока не нашли первую `b`, `q2` если на ленте нет `б`, то мы удаляем первый символ. В `q7` если `b` последняя, то она и удаляется. В `q9` если `б` предпоследняя, то удаляем последний символ. И с `q10` в других случаях и до конца `states`.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	basac	bac	-
2.	acab	aca	-
3.	acaba	acab	-

Выводы

Были изучены базовые принципы работы машины Тьюринга. Была реализована машина Тьюринга на языке Python.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb3.py

```

R=1
L=-1
N=0
states = {'q0': {'a': ('a', R, 'q1'), 'b': ('b', R, 'q6'), 'c': ('c', R,
'q1'), ' ': (' ', R, 'q0')},
          'q1': {'a': ('a', R, 'q1'), 'b': ('b', R, 'q6'), 'c': ('c', R,
'q1'), ' ': (' ', L, 'q2')},
          'q2': {'a': ('a', L, 'q2'), 'c': ('c', L, 'q2'), ' ': (' ', R,
'q3')},
          'q3': {'a': (' ', R, 'q4'), 'c': (' ', R, 'q4'), ' ': (' ', R,
'q4')},
          'q4': {'a': (' ', L, 'q5a'), 'c': (' ', L, 'q5c'), ' ': (' ',
R, 'qend')},
          'q5a': {' ': ('a', R, 'q3')},
          'q5c': {' ': ('c', R, 'q3')},
          'q6': {'a': (' ', R, 'q8'), 'b': (' ', R, 'q8'), 'c': (' ', R,
'q8'), ' ': (' ', L, 'q7')},
          'q7': {'b': (' ', R, 'qend')},
          'q8': {'a': ('a', N, 'q10'), 'b': ('b', N, 'q10'), 'c': ('c',
N, 'q10'), ' ': (' ', L, 'q9')},
          'q9': {'a': (' ', R, 'qend'), 'b': (' ', R, 'qend'), 'c': (' ',
R, 'qend'), ' ': (' ', R, 'qend')},
          'q10': {'a': (' ', R, 'q11'), 'b': (' ', R, 'q11'), 'c': (' ',
R, 'q11'), ' ': (' ', R, 'q11')},
          'q11': {'a': (' ', L, 'q12a'), 'b': (' ', L, 'q12b'), 'c': ('
', L, 'q12c'), ' ': (' ', L, 'q12.2')},
          'q12a': {' ': ('a', R, 'q10')},
          'q12b': {' ': ('b', R, 'q10')},
          'q12c': {' ': ('c', R, 'q10')},
          'q12.2': {' ': (' ', L, 'q13')},
          'q13': {'a': ('a', L, 'q13'), 'b': ('b', L, 'q13'), 'c': ('c',
L, 'q13'), ' ': (' ', N, 'q14')},
          'q14': {'a': (' ', R, 'q15'), 'b': (' ', R, 'q15'), 'c': (' ',
R, 'q15'), ' ': (' ', R, 'q15')},
          'q15': {'a': (' ', L, 'q16a'), 'b': (' ', L, 'q16b'), 'c': ('
', L, 'q16c'), ' ': (' ', R, 'qend')},
          'q16a': {' ': ('a', R, 'q14')},
          'q16b': {' ': ('b', R, 'q14')},
          'q16c': {' ': ('c', R, 'q14')}}
}
lenta = list(input())
state = "q0"
index = 0
while state != "qend":
    symbol, step, state = states[state][lenta[index]]
    lenta[index] = symbol
    index += step

print("".join(lenta))

```