

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

Яковлева А.А.

Преподаватель

Иванов Д.В.

Санкт-Петербург
2023

Цель работы

Изучение работы конечных автоматов, в частности Машины Тьюринга.

Задание

Вариант 4

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}, которая начинается с символа 'a'.

			a	c	c	a	b	c	b	a	b	a	a	c	a	b			
--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

Напишите программу, которая оборачивает исходную строку. Результат работы алгоритма - исходная последовательность символов в обратном порядке.

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:

			b	a	c	a	a	b	a	b	c	b	a	c	c	a			
--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

Алфавит (можно расширять при необходимости):

- 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.
6. Нельзя использовать дополнительную ленту, в которую записывается результат.

Ваша программа должна вывести полученную ленту после завершения работы. В отчет включите таблицу состояний. Отдельно кратко опишите каждое состояние.

Выполнение работы

Таблица состояний:

	a	b	c	'	d
q1	a;L;q2	b;L;q2	c;L;q2	' ;R;q1	
q2				d;R;q3	
q3	a;R;q3	b;R;q3	c;R;q3	' ;L;q4	
q4	a;L;q5	b;L;q5	c;L;q5		
q5	' ;R;q10	' ;R;q11	' ;R;q12	' ;L;q5	' ;n;qT
q6	a;R;q6	b;R;q6	c;R;q6	a;L;q9	
q7	a;R;q6	b;R;q6	c;R;q6	b;L;q9	
q8	a;R;q6	b;R;q6	c;R;q6	c;L;q9	
q9	a;L;q9	b;L;q9	c;L;q9	' ;L;q5	
q10	a;N;q6	b;N;q6	c;N;q6	' ;R;q10	
q11	a;N;q7	b;N;q7	c;N;q7	' ;R;q11	
q12	a;N;q8	b;N;q8	c;N;q8	' ;R;q12	

Рассмотрим каждое состояние:

- *q1* начальное состояние: пока встречается пробел перемещает курсор вправо, когда встречается a, b, или c перемещает курсор влево, на пробел до последовательности символов, и переходит в состояние *q2*
- *q2* вместо пробела записывает d, добавленный в алфавит символ, необходимый для определения начала строки, перемещает курсор вправо и переходит в состояние *q3*
- *q3* пока встречается a, b, или c перемещает курсор вправо, когда встречается пробел перемещает курсор влево, на последний символ последовательности, и переходит в состояние *q4*
- *q4* перемещает курсор влево, на предпоследний символ последовательности, чтобы обернуть последовательность относительно последнего символа, переходит в состояние *q5*
- *q5* перемещает курсор влево пока встречается пробел, когда встречается непробельный символ заменяет его пробелом, если текущий символ был d,

значит курсор вернулся в начало строки, переходит в конечное состояние qT , иначе перемещает курсор вправо и переходит в состояние $q10$ если это a , $q11$ если b , $q12$ если c

- $q6$ перемещает курсор вправо пока встречается a , b или c , когда встречается пробел (конец последовательности в обратном порядке) заменяет его символом a , перемещает курсор влево, переходит в состояние $q9$
- $q7$ перемещает курсор вправо пока встречается a , b или c , когда встречается пробел (конец последовательности в обратном порядке) заменяет его символом b , перемещает курсор влево, переходит в состояние $q9$
- $q8$ перемещает курсор вправо пока встречается a , b или c , когда встречается пробел (конец последовательности в обратном порядке) заменяет его символом c , перемещает курсор влево, переходит в состояние $q9$
- $q9$ перемещает курсор влево пока встречается a , b или c , когда встречается пробел (ближайший пробел к началу последовательности в обратном порядке) перемещает курсор влево, переходит в состояние $q5$
- $q10$ перемещает курсор вправо пока встречается пробел, когда встречается другой символ (начало последовательности в обратном порядке) переходит в состояние $q6$
- $q11$ перемещает курсор вправо пока встречается пробел, когда встречается другой символ (начало последовательности в обратном порядке) переходит в состояние $q7$
- $q12$ перемещает курсор вправо пока встречается пробел, когда встречается другой символ (начало последовательности в обратном порядке) переходит в состояние $q8$

В словарь *table* запишем таблицу состояний. Считаем ленту, так как она бесконечна, добавим к ней справа *tape_length* пробелов, в данной задаче длина строки не более 13, поэтому 20 пробелов будет достаточно, запишем в *memory* список, содержащий элементы полученной ленты. В список состояний q добавим начальное состояние $q1$, idx , указателю на текущую ячейку ленты, присвоим 0.

С помощью цикла *while* пока текущее состояние не равно конечному, *symbol* присвоим символ, стоящий в текущей ячейке, q_curr присвоим текущее

состояние, поменяем символ в текущей ячейке ленты в соответствии с таблицей, в список состояний q добавим соответствующее состояние, idx добавим значение функции $move$ (-1, если в таблице стоит L; 0, если N; 1, если R).

После конечного состояния выведем ленту.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	a	a	
2.	abc	cba	
3.	aabbcc	ccbbaa	
4.	abbcaababccac	caccbabaacbba	

Выводы

Была изучена работа конечных автоматов, создана программа, моделирующая работу Машины Тьюринга.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
def move(direction):
    if direction == 'L':
        return -1
    elif direction == 'N':
        return 0
    else:
        return 1

table = {
    'q1': {'a': ('a', 'L', 'q2'), 'b': ('b', 'L', 'q2'), 'c': ('c',
'L', 'q2'), ' ': (' ', 'R', 'q1')},
    'q2': {' ': ('d', 'R', 'q3')},
    'q3': {'a': ('a', 'R', 'q3'), 'b': ('b', 'R', 'q3'), 'c': ('c',
'R', 'q3'), ' ': (' ', 'L', 'q4')},
    'q4': {'a': ('a', 'L', 'q5'), 'b': ('b', 'L', 'q5'), 'c': ('c',
'L', 'q5')},
    'q5': {'a': (' ', 'R', 'q10'), 'b': (' ', 'R', 'q11'), 'c': ('
', 'R', 'q12'), ' ': (' ', 'L', 'q5'), 'd': (' ', 'N', 'qT')},
    'q6': {'a': ('a', 'R', 'q6'), 'b': ('b', 'R', 'q6'), 'c': ('c',
'R', 'q6'), ' ': ('a', 'L', 'q9')},
    'q7': {'a': ('a', 'R', 'q7'), 'b': ('b', 'R', 'q7'), 'c': ('c',
'R', 'q7'), ' ': ('b', 'L', 'q9')},
    'q8': {'a': ('a', 'R', 'q8'), 'b': ('b', 'R', 'q8'), 'c': ('c',
'R', 'q8'), ' ': ('c', 'L', 'q9')},
    'q9': {'a': ('a', 'L', 'q9'), 'b': ('b', 'L', 'q9'), 'c': ('c',
'L', 'q9'), ' ': (' ', 'L', 'q5')},
    'q10': {'a': ('a', 'N', 'q6'), 'b': ('b', 'N', 'q6'), 'c': ('c',
'N', 'q6'), ' ': (' ', 'R', 'q10')},
    'q11': {'a': ('a', 'N', 'q7'), 'b': ('b', 'N', 'q7'), 'c': ('c',
'N', 'q7'), ' ': (' ', 'R', 'q11')},
    'q12': {'a': ('a', 'N', 'q8'), 'b': ('b', 'N', 'q8'), 'c': ('c',
'N', 'q8'), ' ': (' ', 'R', 'q12')},
}

tape_length = 20
memory = list(input() + " "*tape_length)
q = ['q1']
idx = 0

while q[-1] != 'qT':
    symbol = memory[idx]
    q_curr = q[-1]
    memory[idx] = table[q_curr][symbol][0]
    idx += move(table[q_curr][symbol][1])
    q.append(table[q_curr][symbol][2])

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