

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Введение в информационные технологии»
Тема: Моделирование работы Машины Тьюринга

Студент гр. 9304

Мохаммед.А.А.

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2019

Цель работы.

Ознакомиться с принципом работы Машины Тьюринга и научиться моделировать ее работу на языке Python.

Задание.

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

На ленте находится троичное число и знак(плюс или минус) и троичная цифра.

Написать программу, которая выполнит арифметическую операцию. Указатель на текущее состояние Машины Тьюринга изначально находится слева от числа (но не на первом его символе). По обе стороны от числа находятся пробелы. Результат арифметической операции запишите на месте первого числа.

Алфавит:

0

1

2

+

-

" " (пробел)

Соглашения:

1. Направление движения автомата может быть одно из R (направо), L (налево), N (неподвижно).
2. Число обязательно начинается с единицы или двойки.
3. Числа и знак операции между ними идут непрерывно.
4. Гарантируется, что в результате операции вычитания не может получиться отрицательного числа.

Выполнение работы.

Составляем таблицу состояний. После этого программируем получившуюся таблицу. Каждую строку таблицы (каждое состояние) выделяем в отдельную функцию с соответствующим названием (q_1 , q_2 , q_3 и т.д.). Далее считываем выражение и помещаем каждый символ в список как отдельный элемент. Создаем словарь состояний, каждый ключ которого – состояние, а значение каждого ключа – название соответствующей функции. Начальное состояние – q_0 . После этого, с помощью цикла `while` (условие – пока состояние не станет q_n), вызываем функции для каждого состояния, одно временно при сдвигая переменную i значение 0, 1 или -1, для того, чтобы передвигаться по списку `tape`. Когда значение переменной `state` достигает q_n , завершаем цикл, а на выходе получаем измененный список `tape`. Выводим список, приводя его к строке.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	1-1	0-1
2.	1000-2	0221-2
3.	121+2	200+2
4.	11-2	02-2

Выводы.

Была изучена модель абстрактного вычислителя — Машина Тьюринга. Реализована программа моделирующая работу Машины Тьюринга. Каждое состояние было выделено в отдельную функцию, для вывода списка в виде строки использовался метод строк join.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
def q0(symbol):
    if symbol == '0': return
        '0', 1, 'q0'
    if symbol == '1': return
        '1', 1, 'q0'
    if symbol == '2': return
        '2', 1, 'q0'
    if symbol == '+': return
        '+', 1, 'q1'
    if symbol == '-': return
        '-', 1, 'q2'
    if symbol == ' ': return
        ' ', 1, 'q0'

def q1(symbol):
    if symbol == '0': return
        '0', 0, 'qn'
    if symbol == '1':
        return '1', -1, 'q3' if
symbol == '2':
        return '2', -1, 'q4'

def q2(symbol):
    if symbol == '0': return
        '0', 0, 'qn'
    if symbol == '1':
        return '1', -1, 'q5' if
symbol == '2':
        return '2', -1, 'q6'

def q3(symbol):
    if symbol == ' ': return
        '1', 0, 'qn'
    if symbol == '0': return
        '1', 0, 'qn'
    if symbol == '1': return
        '2', 0, 'qn'
    if symbol == '2':
        return '0', -1, 'q3' if
symbol == '+':
        return '+', -1, 'q3'

def q4(symbol):
    if symbol == '0': return
        '2', 0, 'qn'
    if symbol == '1':
        return '0', -1, 'q3' if
```

```

symbol == '2':
return '1', -1, 'q3' if
symbol == '+':
    return '+', -1, 'q4'

def q5(symbol):
    if symbol == '0':
        return '2', -1, 'q5' if
    symbol == '1':
        return '0', 0, 'q7' if
    symbol == '2':
        return '1', 0, 'qn' if
    symbol == '-':
        return '-', -1, 'q5'

def q6(symbol):
    if symbol == '0':
        return '1', -1, 'q5' if
    symbol == '1':
        return '2', -1, 'q5' if
    symbol == '2':
        return '0', 0, 'qn' if
    symbol == '-':
        return '-', -1, 'q6'

def q7(symbol):
    if symbol == '0':
        return '0', -1, 'q7' if
    symbol == '1':
        return '1', -1, 'q7' if
    symbol == '2':
        return '2', -1, 'q7' if
    symbol == '':
        return ' ', 1, 'q8'

def q8(symbol):
    if symbol == '0': return
        ' ', 1, 'q8'
    if symbol == '1': return
        '1', 0, 'qn'
    if symbol == '2': return
        '2', 0, 'qn'
    if symbol == '+':
        return '+', -1, 'q8' if
    symbol == '-':
        return '-', -1, 'q8' if
    symbol == '':
        return '0', 0, 'qn'

tape = list(input())
dictionary={'q0':q0,'q1':q1,'q2':q2,'q3':q3,'q4':q4,'q5': q5, 'q6': q6,
'q7': q7, 'q8':q8}

```

```
state = 'q0'
i = 0
while state
  != 'qn':
    new_symbol, direction, state = dictionary[state](tape[i])
    tape[i] = new_symbol
i += direction print(''.join(tape))
```

ПРИЛОЖЕНИЕ ТАБЛИЦА СОСТОЯНИЙ

Таблица 2 - Таблица состояний

	0	1	2	+	-	« »
q0	0, q0, R	1, q0, R	1, q0, R	+, q1, R	-, q2, R	‘ ‘, q0, N
q1	0, qn, N	1, q3, L	2, q4, L			
q2	0, qn, N	1, q5, L	2, q6, L			
q3	1, qn, N	2, qn, N	0, q3, L			1, qn, N
q4	2, qn, N	0, q3, L	1, q3, L			
q5	2, q5, L	0, q7, N	1, q7, N			
q6	1, q5, L	2, q5, L	0, qn, N			
q7	0, q7, L	1, q7, L	2, q7, L			‘ ‘, qn, R
q8 (qn)	‘ ‘, q8, N	1, qn, N	2, qn, N	‘+’, qn, L	‘-’, q8, L	0, qn, N

q0 – ищем знак + или –.

q1 – если нашелся плюс. (считываем символ после + и прибавляем, если нашелся 0 после знака операции)

q2 – если нашелся минус. (считываем символ после - и вычитаем, если нашелся 0 после знака операции)

q3 – прибавляем 1.

q4 – прибавляем 2.

q5 – вычитаем 1.

q6 – вычитаем 2.

q7 – проверка на наличие 0.

q8 – конечное состояние.