

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

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Информатика»**  
**Тема: Моделирование работы Машины Тьюринга**

Студент гр. 0382

Гудов Н.Р.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

## **Цель работы.**

Смоделировать работу Машины Тьюринга на языке Python

## **Задание.**

На вход программе подается строка неизвестной длины. Каждый элемент является значением в ячейке памяти ленты Машины Тьюринга. На ленте находится троичное число, знак (плюс или минус) и троичная цифра. Напишите 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. Программа должна вывести полученную ленту после завершения работы.

## **Основные теоретические положения.**

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

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

2) Алфавит ленты - конечное множество всех возможных символов ленты  $\{0, 1, 2, +, -, '\}$ .

3) 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 3 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.

Состояния представлены в табл. 1

Таблица 1 – Состояния автомата

	0	1	2	+	-	' '
q0	0,R,q0	1,R,q0	2,R,q0	+,R,q1	-,R,q2	' ',R,q0
q1	0,N,qT	1,L,q3	2,L,q5			
q2	0,N,qT	1,L,q4	2,L,q6			
q3	1,N,qT	2,N,qT	0,L,q3	+,L,q3		0,N,q3
q4	2,L,q4	0,L,q7	1,N,qT		-,L,q4	
q5	2,N,qT	0,L,q3	1,L,q3	+,L,q5		0,N,q3
q6	1,L,q4	2,L,q4	2,L,qT		-,L,q6	
q7	0,L,q7	1,L,qT	2,L,qT			' ',R,q8
q8	' ',R,q8	1,N,qT	2,N,qT		-,L,q8	0,N,qT

Описание состояний:

q0 – начальное состояние, автомат ищет знак арифметической операции

q1 – состояние определения цифры для суммирования

q2 – состояние определения цифры для вычитания

q3 – состояние обработки числа при прибавлении единицы

q4 – состояние обработки числа при вычитании единицы

q5 – состояние обработки числа при прибавлении двойки

q6 – состояние обработки числа при вычитании двойки

q7 – состояние перехода к левому символу числа

q8 – состояние замены незначащих нулей

Используемые переменные:

Таблица состояний реализована с использованием словаря *table*.

Входная строка преобразуется в список, после чего на нее ссылается имя *memory*.

Состояние сохраняется в переменную *q* и после этого отслеживается по ее значению.

Переменная *symbol* содержит символ, готовый к записи в ленту.

Переменная *index* выполняет соответствующую роль.

Переменная *state* хранит состояние до передачи его переменной *q*.

### Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	2+1	10+1	Ошибок не обнаружено
2.	111-1	110-1	Ошибок не обнаружено
3.	112+2	121+2	Ошибок не обнаружено
4.	11-2	2-2	Ошибок не обнаружено

### Вывод.

В ходе работы был изучен принцип работы машины Тьюринга, была смоделирована работа машины Тьюринга на языке Python. Программа выполняет сложение троичного числа и троичной цифры и записывает результат на месте первого числа. После результат выводится на экран.

Обработка данных происходила с помощью цикла *while*.

Таблица состояний хранится в двумерном словаре, строка в виде списка.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab\_2.py

```
table = {'q0': {'0': ['0', 1, 'q0'], '1': ['1', 1, 'q0'],
               '2': ['2', 1, 'q0'], '+' : ['+', 1, 'q1'],
               '-' : ['- ', 1, 'q2'], ' ': [' ', 1, 'q0']
          },
         'q1': {'0': ['0', 0, 'qT'],
               '1': ['1', -1, 'q3'],
               '2': ['2', -1, 'q5']
          },
         'q2': {'0': ['0', 0, 'qT'],
               '1': ['1', -1, 'q4'],
               '2': ['2', -1, 'q6']
          },
         'q3': {'0': ['1', 0, 'qT'], '1': ['2', 0, 'qT'],
               '2': ['0', -1, 'q3'], '+' : ['+', -1, 'q3'],
               ' ': ['0', 0, 'q3']
          },
         'q4': {'0': ['2', -1, 'q4'], '1': ['0', -1, 'q7'],
               '2': ['1', 0, 'qT'], '-' : ['- ', -1, 'q4']
          },
         'q5': {'0': ['2', 0, 'qT'], '1': ['0', -1, 'q3'],
               '2': ['1', -1, 'q3'], '+' : ['+', -1, 'q5'],
               ' ': ['0', 0, 'q3']
          },
         'q6': {'0': ['1', -1, 'q4'], '1': ['2', -1, 'q4'],
               '2': ['0', -1, 'q7'], '-' : ['- ', -1, 'q6']
          },
         'q7': {'0': ['0', -1, 'q7'], '1': ['1', -1, 'qT'],
               '2': ['2', -1, 'qT'], ' ': [' ', 1, 'q8']
          },
         'q8': {'0': [' ', 1, 'q8'], '1': ['1', 0, 'qT'],
               '2': ['2', 0, 'qT'],
               '-' : ['- ', -1, 'q8'], ' ': ['0', 0, 'qT']
          }
}
```

```
memory = list(input())
q = 'q0'
index = 0

while q != 'qT':
    symbol, delta, state = table[q][memory[index]]
    memory[index] = symbol
    index += delta
    q=state
print(''.join(memory))
```