

**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

Афанасьев Н. С.

Преподаватель

Шевская А. И.

Санкт-Петербург

2020

### Цель работы.

Изучить принцип работы машины Тьюринга, использовать этот алгоритм для решения задач.

### 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.

### Выполнение работы.

В начале программа создаёт словарь *table* со вложенными словарями массивов – таблицу состояний машины Тьюринга.

Таблица 1 – Состояния машины Тьюринга

	'0'	'1'	'2'	' '	'+'	'-'
q0		['1', 1, q1]	['2', 1, q1]	[' ', 1, q0]		
q1	['0', 1, q1]	['1', 1, q1]	['2', 1, q1]	[' ', -1, q2]	['+', 1, q1]	['-', 1, q1]
q2	['0', 0, qT]	['1', -1, q3]	['2', -1, q4]			
q3					['+', -1, q5]	['-', -1, q6]
q4					['+', -1, q7]	['-', -1, q8]
q5	['1', 0, qT]	['2', 0, qT]	['0', -1, q5]	['1', 0, qT]		
q6	['2', -1, q6]	['0', -1, q9]	['1', 0, qT]			
q7	['2', 0, qT]	['0', -1, q5]	['1', -1, q5]			
q8	['1', -1, q6]	['2', -1, q6]	['0', 0, qT]			
q9	['0', 0, qT]	['1', 0, qT]	['2', 0, qT]	[' ', 1, q10]		
q10	[' ', 1, q10]	['1', 0, qT]	[2, 0, qT]	['0', 0, qT]		['-', -1, q10]

- **q0** – начальное состояние, ищет левую границу числа.
- **q1** – читает всю строку, до правой границы.
- **q2** – считает правую цифру (второе слагаемое): если это число – 0, то работа завершается, так как +0 и -0 ничего не меняют; если это число – 1, то переходит к q3; если 2, то к q4
- **q3** и **q4** – читают знак операции для соответствующего слагаемого, в зависимости от чего переходят к q5-q8.
- **q5** – +1 или перенос в старший разряд.
- **q6** – -1 или перенос из старшего разряда. Если получается 0, то переходит к q9.
- **q7** – +2. Если требуется перенос, переходят к q5.
- **q8** – -2. Если требуется перенос из старшего разряда, переходят к q6.
- **q9** – если 0 получилось в начале числа, то переходит к q10.
- **q10** – заменяет 0 на пробел. Если дальше идёт знак “-“, то ставит перед ним 0 (только в случае 1-1 => 0-1).

Далее считывается входная строка (лента) *line*, создаётся переменная *q* для хранения текущей операции (изначально ‘q0’), переменная *curr* для хранения текущего индекса.

Далее создаётся жизненный цикл программы, который заканчивается, когда достигается завершающее состояние. В цикле из словаря считывается новый символ *char*, шаг *step* и следующая операция *qnext*. Текущий символ на ленте заменяется на *char*, индекс меняется на *step*, следующей операцией становится *qnext*.

В конце выводится полученная лента.

Разработанный программный код см. в приложении А.

## **Тестирование.**

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	111+1	112+1	Верно
2.	1-1	0-1	Верно
3.	1000-2	221-2	Верно
4	221+2	1000+2	Верно

## **Выводы.**

Был изучен принцип работы машины Тьюринга. Этот алгоритм был использован для решения поставленной задачи.

Разработана программа, которая создаёт таблицу состояний для машины Тьюринга, считывает исходную строку (ленту), проводит соответствующие таблице операции (сложение или вычитание троичного числа и цифры) и выводит полученную строку.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
table = {
    'q0': {'1': ['1', 1, 'q1'], '2': ['2', 1, 'q1'], ' ': [' ', 1, 'q0']},
    'q1': {'0': ['0', 1, 'q1'], '1': ['1', 1, 'q1'], '2': ['2', 1, 'q1'], ' ':
[' ', -1, 'q2'], '+': ['+', 1, 'q1'], '-': ['- ', 1, 'q1']},
    'q2': {'0': ['0', 0, 'qT'], '1': ['1', -1, 'q3'], '2': ['2', -1, 'q4']},
    'q3': {'+': ['+', -1, 'q5'], '-': ['- ', -1, 'q6']},
    'q4': {'+': ['+', -1, 'q7'], '-': ['- ', -1, 'q8']},
    'q5': {'0': ['1', 0, 'qT'], '1': ['2', 0, 'qT'], '2': ['0', -1, 'q5'], ' ':
['1', 0, 'qT']},
    'q6': {'0': ['2', -1, 'q6'], '1': ['0', -1, 'q9'], '2': ['1', 0, 'qT']},
    'q7': {'0': ['2', 0, 'qT'], '1': ['0', -1, 'q5'], '2': ['1', -1, 'q5']},
    'q8': {'0': ['1', -1, 'q6'], '1': ['2', -1, 'q6'], '2': ['0', 0, 'qT']},
    'q9': {'0': ['0', 0, 'qT'], '1': ['1', 0, 'qT'], '2': ['2', 0, 'qT'], ' ':
[' ', 1, 'q10']},
    'q10': {'0': [' ', 1, 'q10'], '1': ['1', 0, 'qT'], '2': ['2', 0, 'qT'], '
': ['0', 0, 'qT'], '-': ['- ', -1, 'q10']}
}

line = list(input())
q = 'q0'
curr = 0

while q!='qT':
    char, step, qnext = table[q][line[curr]]
    line[curr] = char
    curr += step
    q = qnext

print(''.join(line))
```