

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

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

Студент гр. 0382

Павлов С.Р

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

### **Цель работы.**

Освоение и моделирование работы машины Тьюринга.

### **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.

Ваша программа должна вывести полученную ленту после завершения работы.

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.

### **Основные теоретические положения.**

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.

Алфавит ленты – конечное множество всех возможных символов ленты.

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 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.

Таблица 1 – Таблица состояний

«Элемент»/ Состояние	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10
«0»	„0“,R,q1	„0“,N, STOP“	„1“,N,q 1	„2“,N, STOP	„0“,N, „STOP“	„2“,L,q6	„1“,L,q6	„0“,L,q8	„0“,R,q9	„“,N, „STOP“
«1»	„1“,R,q1	„1“,L,q3	„2“,N, STOP“	„0“,L,q3	„1“,L,q6	„0“,N,q8	„2“,L,q6	„1“,L,q8	„1“,L,q1 0	
«2»	„2“,R,q1	„2“,L,q4	„0“,L,q3	„1“,L,q3	„2“,L,q7	„1“,N,q8	„0“,N,q8	„2“,N,q8	„2“,L,q1 0	
«+»	„+“,R,q2		„+“,L,q1	„+“,L,q4				„+“,L,q8	„+“,L,,S TOP“	
«-»	„-“,R,q5					„-“,L,q6	„-“,L,q7	„-“,L,q8	„-“,L,,S TOP“	
“ ” (Пробел)	„“,R,q1		„1“,R,“S TOP“					„“,N,q9	„“,R,q9	„“,N, „STOP“

И также состояние «STOP», которое является индикатором конца основного цикла, и прекращения реализации алгоритма.

### Описание состояний:

- q1* – начальное состояние, в котором происходит поиск знака.
- q2* – состояние, в которой определяется цифра для операции сложения.
- q3* – состояние, в котором происходит прибавление единицы к троичному числу.

$q4$  – состояние, в котором происходит прибавление двойки к троичному числу.

$q5$  – состояние, в котором определяется число для операции вычитания.

$q6$  – состояние, в котором происходит вычитание единицы из троичного числа.

$q7$  – состояние, в котором происходит вычитание двойки из троичного числа.

$q8$  – состояние, в котором происходит перемещение на пробел перед строкой для выявления незначащих нулей.

$q9$  – состояние, в котором происходит проверка, есть ли число после нуля.

$q10$  – состояние, в котором происходит удаление незначащего нуля (замена его на пробел).

*STOP* – конечное состояние, обозначающее конец работы программы.

*input* – строка, поступающая на вход.

*data* – входные данные из словаря (алгоритма) на текущем элементе итерации.

*index* – номер символа на ленте.

*state* – состояние

*mem\_tape* – входные данные смещенные на одну позицию вправо (нулевой элемент = „“).

В двухуровневом словаре *algorithm* записана таблица состояний. При обращении к словарю по двум ключам (состоянию и номеру элемента на ленте) переменной *data* присваиваются соответствующие значения символа на замену, шага на ленте и состояния. Таким образом, в цикле *while* происходит изменение ленты до тех пор, пока не будет достигнуто конечное состояние *STOP*.

Последний цикл, выполняет работу вывода полученных данных на экран.

**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.	112+1	120+1	Верно
2.	120-2	111-2	Верно
3.	10000-2	2221-2	Верно
4.	11111+2	11120+2	Верно
5.	12021-1	12020-1	Верно

### **Выводы.**

Было исследовано моделирование работы машины Тьюринга.

Разработана программа, выполняющая считывание с клавиатуры исходной ленты и изменяющая ее посредством вычитания или прибавления троичной цифры (после знака) к троичному числу(до знака). Выводится измененная лента

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
input = list(input())
state = 'q1'
index = 0

mem_tape = [0]*(len(input)+1)

k = 0
mem_tape[0] = " "
# без этой обработки не работает.
for k in range(1,len(input)+1):
    mem_tape[k] = input[k-1]

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

```

        '1': ['1', -1, 'q8'],
        '2': ['2', -1, 'q8'],
        '+': ['+', -1, 'q8'],
        '-': ['-', -1, 'q8'],
        ' ': [' ', 0, 'q9']},

    'q9': {'0': ['0', 1, 'q9'],
           '1': ['1', -1, 'q10'],
           '2': ['2', -1, 'q10'],
           '+': ['+', -1, 'STOP'],
           '-': ['-', -1, 'STOP'],
           ' ': [' ', 1, 'q9']},

    'q10': {'0': [' ', 0, 'STOP'],
            ' ': [' ', 0, 'STOP']}
    }

while state != "STOP":
    data = algorithm[state][mem_tape[index]]
    mem_tape[index] = data[0]
    index += int(data[1])
    state = data[2]
l = 1

# Исправление вывода. (Возможно программа не проходила проверку
из-за лишнего пробела при вычитании из больших чисел 10000-2)

for k in range(1,len(mem_tape)):
    if (mem_tape[k] != ' '):
        print(mem_tape[k], end='')

```