

**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**

Студентка гр. 0382

Деткова А.С.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

### Цель работы.

Изучить основы архитектуры компьютера, научиться строить логические схемы с помощью вентилях, узнать о формате представления данных на компьютере.

Узнать про Машину Тьюринга и ее построение.

### 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.

		1	2	1	+	2			
--	--	---	---	---	---	---	--	--	--

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. Для примера выше лента будет выглядеть так:

		2	0	0	+	2			
--	--	---	---	---	---	---	--	--	--

Ваша программа должна вывести полученную ленту после завершения работы.

Алфавит:

✓0

✓1

✓2

✓+

✓-

✓" " (пробел)

Соглашения:

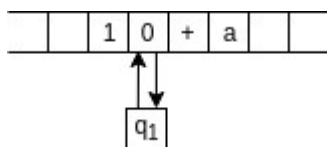
1. Направление движения автомата может быть одно из R (направо), L (налево), N (неподвижно).
2. Число обязательно начинается с единицы или двойки.
3. Числа и знак операции между ними идут непрерывно.
4. Гарантируется, что в результате операции вычитания не может получиться отрицательного числа.

В отчет включите таблицу состояний. Отдельно кратко опишите каждое состояние, например:

q1 - начальное состояние, которое необходимо, чтобы найти первую цифру первого числа. Основные теоретические положения.

Машина Тьюринга

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 именуются. 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. Если предположить, что видимые символы - весь алфавит ленты из примера выше, то мы имеем следующий алфавит:  $\{1, 0, +, 'a', ''\}$ . Последний символ - пустой, означает пустое содержимое клетки.

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 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.

Программа для Машины Тьюринга

Один из способов записи программы для Машины Тьюринга является таблица следующего вида:

	$s_0$	$s_1$	$s_2$	...	$s_m$
$q_0$					
$q_1$					
...					
$q_n$					

$s_0, s_1, s_2, \dots, s_m$  - Алфавит ленты

$q_0, q_1, \dots, q_n$  - Состояния автомата

В ячейках таблицы указываются тройка  $\langle S', [L, R, N], q' \rangle$ :

1.  $S'$  - символ, который необходимо записать в видимую ячейку **ленты**
2.  $[L, R, N]$  - одно из направлений, куда необходимо перейти на **ленте**:

R - направо, L - налево, N - остаться на месте.

3.  $q'$  - состояние, в которое необходимо перейти **автомату**.

## Выполнение работы.

Таблица 1: Таблица состояний

	„0“	„1“	„2“	„+“	„-“	„ „
q0	„0“, +1, „q1“	„1“, +1, „q1“	„2“, +1, „q1“			„ „, +1, „q0“
q1	„0“, +1, „q1“	„1“, +1, „q1“	„2“, +1, „q1“	„+“, +1, „q2“	„-“, +1, „q3“	
q2	„0“, 0, „qt“	„1“, -1, „q4“	„2“, -1, „q5“			
q3	„0“, 0, „qt“	„1“, -1, „q6“	„2“, -1, „q7“			
q4	„1“, 0, „qt“	„2“, 0, „qt“	„0“, -1, „q4“	„+“, -1, „q4“		„1“, 0, „qt“
q5	„2“, 0, „qt“	„0“, -1, „q4“	„1“, -1, „q4“	„+“, -1, „q5“		
q6	„2“, -1, „q6“	„0“, -1, „q8“	„1“, 0, „qt“		„-“, -1, „q6“	
q7	„1“, -1, „q6“	„2“, -1, „q6“	„0“, -1, „q8“		„-“, -1, „q7“	
q8	„0“, 0, „qt“	„1“, 0, „qt“	„2“, 0, „qt“			„ „, +1, „q9“
q9	„ „, +1, „q9“	„1“, 0, „qt“	„2“, 0, „qt“		„-“, -1, „q9“	„0“, 0, „qt“

### Описание состояний:

q0 — начальное состояние, которое необходимо, чтобы найти первую цифру первого числа.

q1 — состояние, которое необходимо, чтобы узнать какое арифметическое действие будет выполняться с числами.

q2 — состояние, которое описывает, какие действия выполняются при каждом из значений второго числа (при сложении).

q3 — состояние, которое описывает, какие действия выполняются при каждом из значений второго числа (при вычитании).

q4 — состояние, которое описывает действия при сложении первого числа с единицей.

q5 — состояние, которое описывает действия при сложении первого числа с двойкой.

q6 — состояние, которое описывает действия при вычитании единицы из первого числа.

q7 — состояние, которое описывает действия при вычитании двойки из первого числа.

q8 — состояние, которое при нулевых значениях последних разрядов переводит указатель индекса на самый первый пробел слева от первого числа.

q9 — состояние, которое превращает первые незначащие нули числа в пробелы.

Разработанный программный код см. в приложении А.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 2: Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	1-1	0-1	Корректная работа программы.
2.	101-2	22-2	Корректная работа программы.
3.	100-1	22-1	Корректная работа программы.
4.	222+2	1001	Корректная работа программы.
5.	111-1	110-1	Корректная работа программы.
6.	121+2	200+2	Корректная работа программы.
7.	21-2	12-2	Корректная работа программы.
8.	212+2	220+2	Корректная работа программы.

## Выводы.

Были изучены основы архитектуры компьютера, способы постройки логических схем с помощью вентиляей, узнали о формате представления данных на компьютере.

Изучили Машину Тьюринга и ее построение.

Была разработана программа, выполняющая считывание с клавиатуры строки неизвестной длины. Каждый элемент является значением в ячейке памяти ленты Машины Тьюринга. На ленте находится троичное число, знак (плюс или минус) и троичная цифра. Для обработки данных использовалась написанная машина Тьюринга. Для каждого состояния выполняется свое действие.



## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2.py

```
tape = list(input())
q = 'q0'
index = 0
conditions = {'q0': {'0': ['0', +1, 'q1'],
                     '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, 'q3']}},
              'q2': {'0': ['0', 0, 'qt'],
                     '1': ['1', -1, 'q4'],
                     '2': ['2', -1, 'q5']}},
              'q3': {'0': ['0', 0, 'qt'],
                     '1': ['1', -1, 'q6'],
                     '2': ['2', -1, 'q7']}},
              'q4': {'0': ['1', 0, 'qt'],
                     '1': ['2', 0, 'qt'],
                     '2': ['0', -1, 'q4'],
                     '+': ['+', -1, 'q4'],
                     ' ': ['1', 0, 'qt']}},
              'q5': {'0': ['2', 0, 'qt'],
                     '1': ['0', -1, 'q4'],
                     '2': ['1', -1, 'q4'],
                     '+': ['+', -1, 'q5']}},
              'q6': {'0': ['2', -1, 'q6'],
                     '1': ['0', -1, 'q8'],
                     '2': ['1', 0, 'qt'],
                     '-': ['-', -1, 'q6']}},
              'q7': {'0': ['1', -1, 'q6'],
                     '1': ['2', -1, 'q6'],
                     '2': ['0', -1, 'q8'],
                     '-': ['-', -1, 'q7']}},
              'q8': {'0': ['0', 0, 'qt'],
                     '1': ['1', 0, 'qt'],
                     '2': ['2', 0, 'qt'],
                     ' ': [' ', +1, 'q9']}},
              'q9': {'0': [' ', +1, 'q9'],
                     '1': ['1', 0, 'qt'],
                     '2': ['2', 0, 'qt'],
                     '-': ['-', -1, 'q9'],
                     ' ': ['0', 0, 'qt']}},
              }
while q != 'qt':
    tape[index], delta, q = conditions[q][tape[index]]
    index += delta
print(''.join(tape))
```