

**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

\_\_\_\_\_

Куликов М.Д.

Преподаватель

\_\_\_\_\_

Шевская Н.В.

Санкт-Петербург

2020

### **Цель работы.**

Ознакомление с принципами работы Машины Тьюринга и самостоятельное ее моделирование с помощью языка программирования Python.

### **Задание.**

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.

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.**

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

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.

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.

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.

В данной работе были использованы такие конструкции языка Python:

`print()` - функция для вывода в консоль

`input()` - считывание данных с консоли, возвращает строку

`.join` — метод, который возвращает строку, собранную из элементов указанного объекта, поддерживающего итерирование.

`while` - цикл, в котором его тело выполняется до тех пор, пока условие цикла истинно

## **Выполнение работы.**

Для выполнения поставленной задачи был разработан алгоритм, соответствующий работе Машины Тьюринга и была составлена таблица всех состояний, которые используются в этом алгоритме.

Код программы был реализован с помощью вложенных словарей, со вложенными в них списками, в которых были описаны указания работы для определенного состояния МТ.

В начале программы с помощью функций `list` и `input` мы вводим данные и приводим их к типу списка.

В конце программы мы выполняем цикл `while`, в котором выполняются действия в соответствии с текущим состоянием до тех пор, пока не будет достигнуто состояние `qT`.

Далее приведена таблица состояний МТ с описанием каждого состояния:

- q1 – проходятся все пробелы и ищется начало числа
- q2 – поиск пробела справа
- q3 – определяется, какая цифра будет вычитаться или складываться
- q4 – состояние для операции с единицей
- q5 – состояние для операции с двойкой
- q6 – состояние для прибавления единицы
- q7 – состояние для вычета единицы
- q8 – состояние для прибавления двойки
- q9 – состояние для вычета двойки
- q10 – состояние для прибавления единицы в старший разряд
- q11 – состояние для вычета единицы из старшего разряда
- q12 – состояние для удаления незначащих нулей
- q13 – добавления нуля для случая, если результатом вычитания оказался ноль
- qT – конечное состояние

Таблица состояний представлена в таблице 1.

Таблица 1 — Таблица состояний

	0	1	2	+	-	‘ ‘
q1	0,R,q2	1,R,q2	2,R,q2			‘ ‘,R,q1
q2	0,R,q2	1,R,q2	2,R,q2	+,R,q2	-,R,q2	‘ ‘,L,q3
q3	0,N,qT	1,L,q4	2,L,q5			
q4				+,L,q6	-,L,q7	
q5				+,L,q8	-,L,q9	
q6	1,N,qT	2,N,qT	0,L,q10			
q7	2,L,q11	0,N,qT	1,N,qT			
q8	2,N,qT	0,L,q10	1,L,q10			
q9	1,L,q11	2,L,q11	0,N,qT			
q10	1,N,qT	2,N,qT	0,L,q10			1,N,qT
q11	2,L,q11	0,L,q12	1,N,q12			
q12	‘ ‘,R,q12	1,N,qT	2,N,qT	+,N,q13	-,N,q13	‘ ‘,R,q12
q13						0,N,qT

### **Тестирование.**

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	122+1	200+1	Корректное выполнение
2.	100+2	102+2	Корректное выполнение
3.	100-1	22-1	Корректное выполнение
4.	1-1	0-1	Корректное выполнение

### **Выводы.**

В ходе работы были изучены принципы работы Машины Тьюринга и была создана МТ на языке программирования Python, выполняющая нужную задачу.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
memory = list(input())
```

```
index = 0
```

```
state = 'q1'
```

```
table = {'q1': {'1': ['1', 1, 'q2'],  
                '2': ['2', 1, 'q2'],  
                ' ': [' ', 1, 'q1']},
```

```
        'q2': {'0': ['0', 1, 'q2'],  
                '1': ['1', 1, 'q2'],  
                '2': ['2', 1, 'q2'],  
                ' ': [' ', -1, 'q3'],  
                '+': ['+', 1, 'q2'],  
                '-': ['-', 1, 'q2']},
```

```
        'q3': {'0': ['0', 0, 'qT'],  
                '1': ['1', -1, 'q4'],  
                '2': ['2', -1, 'q5'], },
```

```
        'q4': {'+': ['+', -1, 'q6'],  
                '-': ['-', -1, 'q7']},
```

```
        'q5': {'+': ['+', -1, 'q8'],  
                '-': ['-', -1, 'q9']},
```

```
        'q6': {'0': ['1', 0, 'qT'],  
                '1': ['2', 0, 'qT'],  
                '2': ['0', -1, 'q10']},
```

'q7': {'0': ['2', -1, 'q11'],  
      '1': ['0', 0, 'qT'],  
      '2': ['1', 0, 'qT']},

'q8': {'0': ['2', 0, 'qT'],  
      '1': ['0', -1, 'q10'],  
      '2': ['1', -1, 'q10']},

'q9': {'0': ['1', -1, 'q11'],  
      '1': ['2', -1, 'q11'],  
      '2': ['0', 0, 'qT']},

'q10': {'0': ['1', 0, 'qT'],  
       '1': ['2', 0, 'qT'],  
       '2': ['0', -1, 'q10'],  
       ' ': ['1', 0, 'qT']},

'q11': {'0': ['2', -1, 'q11'],  
       '1': ['0', -1, 'q12'],  
       '2': ['1', 0, 'q12']},

'q12': {'0': [' ', 1, 'q12'],  
       '1': ['1', 0, 'qT'],  
       '2': ['2', 0, 'qT'],  
       '+': ['+', -1, 'q13'],  
       '-': ['-', -1, 'q13'],  
       ' ': [' ', 1, 'q12']},

'q13': {' ': ['0', 0, 'qT']}



```
}
```

```
while state != 'qT':
```

```
    sym, delta, state = table[state][memory[index]]
```

```
    memory[index] = sym
```

```
    index += delta
```

```
print(''.join(memory))
```