

**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. 8381

\_\_\_\_\_ Почаев Н.А.

Преподаватель

\_\_\_\_\_ Размочаева Н.В.

Санкт-Петербург

2018

## Цель работы

Изучить логические основы работы компьютера, научиться использовать логические операции. На примере моделирования машины Тьюринга изучить применение побитовых операций для операций над числами, в том числе в системах счисления, отличных от 2-ой и 10-ой.

Реализовать алгоритм машины Тьюринга для сложения/вычитания числа произвольной длины с цифрой.

## 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:

Ваша программа должна вывести полученную ленту после завершения работы.

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

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.

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

### **Выполнение работы**

Написание работы производилось на базе операционной системы Windows 10 в интегрированной среде разработки PyCharm.

Ход выполнения работы:

1. В текстовом редакторе Word была создана таблица с возможными состояниями машины Тьюринга и соответствующие команды для неё. (см. Приложение А).
2. В начале программы происходит считывание списка (исходной ленты type) с клавиатуры. Далее переменной state, отвечающей за текущее состояние машины, присваивается значение q1, а переменной pos, отвечающей за текущую позицию автомата на ленте, нулевое значение. (Исходный код см. Приложение Б).
3. Затем идёт цикл while с условием выхода state = "q12", т.е. состоянии, при котором необходимые операции выполнены и машина завершает свою работу.
4. В теле цикла происходит проверка текущего состояния и последующие операции в зависимости от содержания ячеек. Ветвление осуществляется при помощи условного оператора if-elif-else.
5. После выхода из цикла программа выводит полученный список в виде строки (происходит преобразование), при помощи оператора string.join().
6. Также в данной программе используется переменная res, в которую записываются все выполненные команды. Это может использоваться для отладки и изучения работы алгоритма.

### **Выводы**

В ходе лабораторной работы были изучены основы использования побитовых операций в языке Python и их применение для арифметических операций в различных системах счисления. Были приобретены знания по разработке алгоритма эмулирования работы Тьюринга и реализации их в виде

кода. Также были получены знания по представлению чисел в памяти компьютера и основам низкоуровневых операций с ними на примере абстрактного исполнителя.

## ПРИЛОЖЕНИЕ А. АЛГОРИТМ РАБОТЫ МАШИНЫ ТЬЮРИНГА

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

Под «завершением программы» понимаем переход в состояние q12.

q1 - Перемещение указателя от левого края ленты до знака арифметической операции, переход к состоянию q2 справа от знака.

q2 - Если цифра после знака равна 0, то не совершаем перемещение и завершаем выполнение программы. Если равна 1, то перемещаемся влево и переходим в состояние q3. Если равна 2, то перемещаемся влево и переходим в состояние q8.

q3 - Если знак арифметической операции равен “+”, то переходим в состояние q4. Если знак арифметической операции равен “-”, то переходим в состояние q5. В обоих случаях сдвигаемся влево.

q4 - Если первая (с конца) цифра числа 0, то прибавляем единицу (записываем 1), не совершаем перемещение и завершаем программу. Если равна 1, то прибавляем 1 (записываем 2), не совершаем перемещение и завершаем программу. Если равна 2, то записываем 0 (единицу переносим в старший разряд), сдвигаемся влево и не меняем состояние. Если после переноса знака мы оказываемся в ячейке, содержащей 0, то записываем в неё 1 и завершаем программу.

q5 - Если первая (с конца) цифра равна 0, то записываем в ячейку 1 (занимаем у старшего разряда), сдвигаемся влево и не меняем состояние. Если равна 1, то записываем 0 и переходим в состояние q6 для проверки на незначащие 0 перед

числом - результатом операций (может возникнуть в случае займа у старшего разряда - единицы).

q6 - Перемещение в часть ленты перед числом и проверка на незначащие нули, при встрече пробела переход в состояние q11.

q7 - При отсутствии нулей - завершение программы.

q8 - Определение арифметической операции в случае цифры “2” справа от знака. Переход в состояние q9, если равно + и в q10, если равно -.

q9 - Сложение 2 с цифрой, находящейся в ячейке ленты и запись результата: самой двойки (в случае 0) и остатка от переноса разряда, так же в данном случае переходим в состояние q4 для арифметических операций с переносом разряда.

q10 - Если цифра в ячейке равна 0, то записываем 2 и завершаем программу, в случае с 1 и 2 делаем перенос в старший разряд и переходим в состояние q4. В случае с 1, записываем в ячейку 0, с 2 - 1.

q11 - Заменяем все незначащие нули на пробелы, если результат арифметической операции равен нулю, то отлавливаем этот случай, переходим в q7 и записываем полученный 0. В ином случае завершаем программу.

## ПРИЛОЖЕНИЕ Б. ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл: main.py

```
tape = [i for i in input()]
res = ''
state = "q1"
pos = 0
while state != "q12":
    res += state + " " # выполненные операции для проверки

    if state == "q1": # сдвиг вправо
        if tape[pos] == "+" or tape[pos] == "-":
            state = "q2"
            pos += 1

    elif state == "q2": # определение цифры справа от знака
        if tape[pos] == "0":
            state = "q12"
        elif tape[pos] == "1":
            pos -= 1
            state = "q3"
        elif tape[pos] == "2":
            pos -= 1
            state = "q8"

    elif state == "q3": # определение знака для 1
        if tape[pos] == "+":
            state = "q4"
        elif tape[pos] == "-":
            state = "q5"
        pos -= 1

    elif state == "q4": # сложение для 1
        if tape[pos] == "1" or tape[pos] == "0" or tape[pos] == " ":
            state = "q12"
            if tape[pos] == "0":
                tape[pos] = "1"
            elif tape[pos] == "1":
```

```

        tape[pos] = "2"
    elif tape[pos] == " ":
        tape[pos] = "1"
    elif tape[pos] == "2":
        tape[pos] = "0"
        pos -= 1
    else:
        pos -= 1

elif state == "q5": # вычитание для 1
    if tape[pos] == "0":
        tape[pos] = "2"
        pos -= 1
    elif tape[pos] == "1":
        tape[pos] = "0"
        state = "q6"
        pos -= 1
    elif tape[pos] == "2":
        tape[pos] = "1"
        state = "q12"

elif state == "q6": # переход в начало числа
    if tape[pos] == " ":
        state = "q11"
        pos += 1
    else:
        state = "q12"

elif state == "q7": # вставка результирующего нуля
    tape[pos] = "0"
    state = "q12"

elif state == "q8": # определение знака для 2
    if tape[pos] == "+":
        state = "q9"
    elif tape[pos] == "-":
        state = "q10"
    pos -= 1

```



```

elif state == "q9": # сложение для 2
    if tape[pos] == "0" or tape[pos] == " ":
        tape[pos] = "2"
        state = "q12"
    elif tape[pos] == "1":
        tape[pos] = "0"
        pos -= 1
        state = "q4"
    elif tape[pos] == "2":
        tape[pos] = "1"
        pos -= 1
        state = "q4"
    else:
        pos -= 1

elif state == "q10": # вычитание для 2
    if tape[pos] == "2":
        tape[pos] = "0"
        state = "q6"
    else:
        state = "q5"
        if tape[pos] == "0":
            tape[pos] = "1"
        elif tape[pos] == "1":
            tape[pos] = "2"
        pos -= 1

elif state == "q11": # удаление незначащих 0
    if tape[pos] == "0":
        tape[pos] = " "
        pos += 1
    elif tape[pos] == "+" or tape[pos] == "-":
        state = "q7"
        pos -= 1
    else:
        state = "q12"

```

```
res += "q12"  
print(''.join(tape))
```