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

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Введение в информационные технологии»

Тема: «Моделирование работы Машины Тьюринга»

Студент гр. 9383 _____ Чумак М.А.

Преподаватель _____ Размочаева Н.В.

Санкт-Петербург

2019

Цель работы.

Изучить работу Машины Тьюринга и реализовать алгоритм сложения и вычитания троичного числа и троичной цифры.

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.

Выполнение работы.

Перед написанием программы было необходимо составить таблицу всех возможных вариантов. Таблица имеет вид столбцов в качестве отдельных знаков алфавита и строк в качестве переходных состояний, которые решают ту или иную подзадачу.

В таблице, кроме заполненных ячеек, содержатся так же и пустые. Это связано с тем, что в определённом состоянии автомат не встретит тот или иной знак алфавита, который пересекается с данной ячейкой.

Таблица 1 — Состояния автомата Машины Тьюринга

	0	1	2	+	-	« »
q1	0; R; q1	1; R; q1	2; R; q1	+; R; q2	-; R; q3	« »; R; q1
q2	0; N; q10	1; L; q4	2; L; q5			
q3	0; N; q10	1; L; q6	2; L; q7			
q4	1; N; q10	2; N; q10	0; L; q4	+; L; q4		1; N; q10
q5	2; N; q10	0; L; q4	1; L; q4	+; L; q5		
q6	2; L; q6	0; L; q8	1; N; q10		-; L; q6	
q7	1; L; q6	2; L; q6	0; N; q10		-; L; q7	
q8	0; N; q10	1; N; q10	2; N; q10			« »; R; q9
q9	« »; R; q9	1; N; q10	2; N; q10	+; L; q9	-; L; q9	0; N; q10

Описание состояний:

- q1 — поиск знака + или -
- q2 — прибавление того или иного числа
- q3 — вычитание того или иного числа
- q4 — прибавление 1 к троичному числу
- q5 — прибавление 2 к троичному числу
- q6 — вычитание 1 из троичного числа
- q7 — вычитание 2 из троичного числа
- q8 — поиск символа слева от 0 после вычитания
- q9 — проверка 0 на значимость

Для работы с данной таблицей было необходимо все состояния внести в словарь *dictionary_of_states*, в котором идёт обращение к определённому символу строки через ключ-состояние. В список *tape* поступает строка ввода, по элементам которой можно проходить с помощью переменной *index*,

отвечающей за номер элемента в списке, и *step*, служащий для перемещения влево (-1), вправо (1), либо останавливающийся на том же символе (0). В переменную *state* записываем начальное состояние *q1*.

Цикл *while* продолжает выполняться, пока не будет встречено конечное состояние (q10). В теле цикла производится обращение к словарю состояний при помощи конструкции *dictionary_of_states[«текущее состояние»] [«встреченный символ»]* и происходит поэлементное присваивание.

Выводится итоговая строка после всех нужных нам преобразований.

Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	„ 2+2 “	„ 11+2 “
2.	„ 1-1“	„ 0-1“
3.	„ 100-2 “	„ 21-2 “

Выводы.

Были изучены основные принципы работы Машины Тьюринга. Была написана программа, которая прибавляет или вычитает из троичного числа любую троичную цифру. На языке Python были применены базовые конструкции и словарь для достижения цели данной работы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

Код:

```
dictionary_of_states = {
    'q1': {
        '0': ('0', 1, 'q1'),
        '1': ('1', 1, 'q1'),
        '2': ('2', 1, 'q1'),
        '+': ('+', 1, 'q2'),
        '-': ('-', 1, 'q3'),
        ' ': (' ', 1, 'q1')
    },
    'q2': {
        '0': ('0', 0, 'q10'),
        '1': ('1', -1, 'q4'),
        '2': ('2', -1, 'q5')
    },
    'q3': {
        '0': ('0', 0, 'q10'),
        '1': ('1', -1, 'q6'),
        '2': ('2', -1, 'q7')
    },
    'q4': {
        '0': ('1', 0, 'q10'),
        '1': ('2', 0, 'q10'),
        '2': ('0', -1, 'q4'),
        '+': ('+', -1, 'q4'),
        ' ': ('1', 0, 'q10')
```

```
},
'q5': {
    '0': ('2', 0, 'q10'),
    '1': ('0', -1, 'q4'),
    '2': ('1', -1, 'q4'),
    '+': ('+', -1, 'q5'),
},
'q6': {
    '0': ('2', -1, 'q6'),
    '1': ('0', -1, 'q8'),
    '2': ('1', 0, 'q10'),
    '-': ('-', -1, 'q6')
},
'q7': {
    '0': ('1', -1, 'q6'),
    '1': ('2', -1, 'q6'),
    '2': ('0', 0, 'q10'),
    '-': ('-', -1, 'q7')
},
'q8': {
    '0': ('0', 0, 'q10'),
    '1': ('1', 0, 'q10'),
    '2': ('2', 0, 'q10'),
    ' ': (' ', 1, 'q9')
},
'q9': {
    '0': (' ', 1, 'q9'),
    '1': ('1', 0, 'q10'),
    '2': ('2', 0, 'q10'),
```

```
        '+': ('+', -1, 'q9'),
        '-': ('-', -1, 'q9'),
        ' ': ('0', 0, 'q10'),
    }
}

tape = list(input())
state = 'q1'
index = 0
while state != 'q10':
    tape[index], step, state = dictionary_of_states[state][tape[index]]
    index += step
print(''.join(tape))
```