

**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

Диденко Д.В.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

## Цель работы.

Изучение 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.

		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.

### **Основные теоретические положения.**

Машина Тьюринга состоит из неподвижной *ленты* (аналог памяти в реальной вычислительной машине) и *автомата* (процессора). Лента (память) используется для хранения информации. Она бесконечна в обе стороны и разбита на ячейки, которые никак не нумеруются и не именуются. В каждой клетке может быть либо записан один символ, либо ничего не записано. Все возможные символы, которые могут храниться на ленте, образуют алфавит. Важно отметить, что при рассмотрении машины Тьюринга принимается допущение о том, что алфавит всегда конечен.

Ее функционирование заключается в последовательном перемещении автомата вдоль ленты с возможным (но необязательным) изменением символов, хранящихся в ее ячейках. В каждый момент времени автомат размещается целиком только под одной из клеток ленты (автомат не может находиться между клетками) и может прочитать ее содержимое; содержимое других клеток автомат не видит. Перемещение процессора задается программой – правилами перехода. В процессе работы машины Тьюринга в каждый момент времени автомат находится в одном состоянии, которое обычно обозначается буквой  $q$  с номерами:  $q_0$ ,  $q_1$ ,  $q_2$  и т.д. Правила перехода задает действия, которые автомат должен выполнить, в зависимости от текущего состояния и символа на ленте, а также следующее состояние  $q_n$ , в которое автомату необходимо перейти. Существует конечное состояние (терминальное), в котором автомат останавливается (машина Тьюринг останавливает свою работу). Программа для машины Тьюринга представляется в виде таблицы переходов. Столбцы соответствуют символам

алфавита, а строки – состояниям автомата. В ячейках таблицы указываются тройки <Symbol', [L, R, N], q'>:

- Symbol' – символ, который необходимо записать в видимую ячейку ленты.
- [L, R, N] – одно из направлений, куда нужно перейти на ленте:
  - R - направо,
  - L - налево,
  - N - остаться на месте.
- q' – состояние, в которое необходимо перейти автомату.

### Выполнение работы.

Исходный код решения задачи см.в приложении А.

Состояния представлены в табл.1.

Таблица 1 – Таблица правил перехода

	' '	'0'	'1'	'2'	'+'	'-'
q0	' ',1,'q0'	'0',1,'q0'	'1',1,'q0'	'2',1,'q0'	'+',1,'qX'	'-',1,'qX2'
qX	' ',0,'qT'	'0',0,'qT'	'1',-1,'q1'	'2',-1,'q2'	'+',-1,'qX'	'-',-1,'qX2'
q1	'1',0,'qT'	'1',0,'qT'	'2',0,'qT'	'0',-1,'q1'	'+',-1,'q1'	'-',-1,'q1'
q2	' ',0,'qT'	'2',0,'qT'	'0',-1,'q1'	'1',-1,'q1'	'+',-1,'q2'	'-',-1,'q2'
qX2	-	'0',0,'qT'	'1',-1,'q3_1'	'2',-1,'q4_1'	-	-
q3_1	-	'2',-1,'q3_2'	'0',0,'qT'	'1',0,'qT'	'+',-1,'q3_1'	'-',-1,'q3_1'
q3_2	-	'2',-1,'q3_2'	'0',-1,'qE'	'1',0,'qT'	-	-
q4_1	-	'1',-1,'q3_2'	'2',-1,'q3_2'	'0',0,'qT'	'+',-1,'q4_1'	'-',-1,'q4_1'
qE	' ',1,'qE2'	'0',0,'qT'	'1',0,'qT'	'2',0,'qT'	-	-
qE2	' ',0,'qT'	' ',0,'qT'	'1',0,'qT'	'2',0,'qT'	-	-

q0 - начальное состояние, необходимое, чтобы определить вид алгебраической операции. Двигается по ленте слева направо(фактически не изменяет символы ленты) пока не увидит операнд. Если «+», то машина

переходит к состоянию  $qX$ , если «-» - к состоянию  $qX2$ , при любом алгебраическом знаке смещается на 1 вправо.

$qX$  – состояние, необходимое, чтобы определить слагаемое. Если  $qX$  видит «0», то не смещается и переводит в терминальное состояние, «1» - смещается влево, состояние  $q1$ , «2» - смещается влево, состояние  $q2$ .

$q1$  – состояние, при котором слагаемое равно 1. Смещается влево при любом алгебраическом знаке, остается в нынешнем состоянии. При «0»: заменяет на «1», переводит в терминальное состояние, при «1»: заменяет на «2», переводит в терминальное состояние, при «2»: заменяет на «0», смещается влево, остается в нынешнем состоянии, при « »: заменяет на «1», переводит в терминальное состояние

$q2$  – состояние, при котором слагаемое равно 2. Смещается влево при любом алгебраическом знаке, остается в нынешнем состоянии. При «0»: заменяет на «2», переводит в терминальное состояние, при «1»: заменяет на «0», смещается влево, переводит в состояние  $q1$ , при «2»: заменяет на «1», смещается влево, переводит в состояние  $q1$ .

$qX2$  – состояние, необходимое, чтобы определить вычитаемое. При «0»: переводит в терминальное состояние, при «1» - смещается влево, переводит в состояние  $q3\_1$ , при «2» - смещается влево, переводит в состояние  $q4\_1$ .

$q3\_1$  - состояние, при котором вычитаемое равно 1. При «0»: заменяет на «2», смещается влево, переходит в состояние  $q3\_2$ , при «1»: заменяет на «0», остается на месте, переводит в терминальное состояние, при «2»: заменяет на «1», остается на месте, переводит в терминальное состояние, при любом знаке смещается влево, остается в нынешнем состоянии.

$q3\_2$  – состояние, реализующее вычет разряд, переданный в младший разряд числа. При «0»: заменяет на «2», смещается влево, остается в нынешнем состоянии, при «1»: заменяет на «0», не смещается, переводит в состояние  $qE$ , при «2»: заменяет на «1», не смещается, переводит в терминальное состояние.

$q4\_1$  - состояние, при котором вычитаемое равно 2. При «0»: заменяет на «1», смещается влево, переводит в состояние  $q3\_2$ , при «1»: заменяет на «2»,

смещается влево, переводит в состояние  $q3\_2$ , при «2»: заменяет на «0», остается на месте, переводит в терминальное состояние.

$qE$  - состояние, необходимое, чтобы определить являлся ли разряд самым старшим. Если так, то на ленте встретится « », и автомат сместится вправо и переведет в состояние  $qE2$ . В любом другом случае – переводит в терминальное состояние.

$qE2$  - состояние, необходимое для удаления незначащего нуля. При «0»: заменяет на « », остается на месте, переводит в терминальное состояние. В других случаях переводит в терминальное состояние.

На вход программе подается строка, преобразованная с помощью функции *list()* в список *tem*. Состояния автомата описываются в двухмерном словаре *task*. Первый «уровень» составляют состояние, второй – символ, значение для которого – список, состоящий из команд, которые нужно сделать с текущим состоянием для текущего символа. Каждый список содержит три элемента:

[«меняемый символ»,  
направление смещения(-1 – влево, 0 – остаться на месте, 1 - вправо),  
«новое состояние»].

В переменной *state* будет храниться текущее состояние автомата, сперва записывается начальное положение автомата « $q0$ ». В переменной *index* хранится индекс текущего элемента «ленты», сперва равен 0. В переменной *sum* хранится символ на который заменяется текущий элемент ленты. В цикле *while* происходит моделирование пробежки автомата по ленте, цикл выполняется, пока состояние не равно терминальному. Через функцию *print()* на экран выводится строка, преобразованная из измененного после цикла списка *tem* с помощью метода *join()*.

## Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	10000-2	2221-2	Программа работает верно
2.	1001-2	222-2	Программа работает верно
3.	1-1	0-1	Программа работает верно
4.	2-2	0-2	Программа работает верно
5.	1000-1	222-1	Программа работает верно
6.	100100-1	100022-1	Программа работает верно
7.	101000-2	100221-2	Программа работает верно
8.	2221-2	2212-2	Программа работает верно

### Выводы.

Были изучены принципы работы машины Тьюринга. Смоделирована работа машины Тьюринга.

Разработана программа, которая принимает на вход строку неизвестной длины с записью : «”троичное число”+(-)”троичная цифра”», слева и справа от записи находятся пробелы, и совершает арифметическое действие в зависимости от знака, указанного в строке. Программа работает по модели машины Тьюринга.

# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД ПРОГРАММЫ

**Название файла:** Turing.py

```

mem = list(input())
task = {'q0':    {' ' : [' ', 1, 'q0'],
                  '0': ['0', 1, 'q0'],
                  '1': ['1', 1, 'q0'],
                  '2': ['2', 1, 'q0'],
                  '+': ['+', 1, 'qX'],
                  '-': ['-', 1, 'qX2']}},
'qX':    {' ' : [' ', 0, 'qT'],
          '0': ['0', 0, 'qT'],
          '1': ['1', -1, 'q1'],
          '2': ['2', -1, 'q2'],
          '+': ['+', -1, 'qX'],
          '-': ['-', -1, 'qX2']}},
'q1':    {' ' : ['1', 0, 'qT'],
          '0': ['1', 0, 'qT'],
          '1': ['2', 0, 'qT'],
          '2': ['0', -1, 'q1'],
          '+': ['+', -1, 'q1'],
          '-': ['-', -1, 'q1']}},
'q2':    {' ' : [' ', 0, 'qT'],
          '0': ['2', 0, 'qT'],
          '1': ['0', -1, 'q1'],
          '2': ['1', -1, 'q1'],
          '+': ['+', -1, 'q2'],
          '-': ['-', -1, 'q2']}},
'qX2':   {'0': ['0', 0, 'qT'],
          '1': ['1', -1, 'q3_1'],
          '2': ['2', -1, 'q4_1']}},
'q3_1':  {'0': ['2', -1, 'q3_2'],
          '1': ['0', 0, 'qT'],
          '2': ['1', 0, 'qT'],
          '+': ['+', -1, 'q3_1'],
          '-': ['-', -1, 'q3_1']}},
'q3_2':  {'0': ['2', -1, 'q3_2'],
          '1': ['0', -1, 'qE'],
          '2': ['1', 0, 'qT']}},
'q4_1':  {'0': ['1', -1, 'q3_2'],
          '1': ['2', -1, 'q3_2'],
          '2': ['0', 0, 'qT'],
          '+': ['+', -1, 'q4_1'],
          '-': ['-', -1, 'q4_1']}},
'qE':    {' ' : [' ', 1, 'qE2'],
          '0': ['0', 0, 'qT'],
          '1': ['1', 0, 'qT'],
          '2': ['2', 0, 'qT']}},
'qE2':   {' ' : [' ', 0, 'qT'],
          '0': [' ', 0, 'qT'],
          '1': ['1', 0, 'qT'],
          '2': ['2', 0, 'qT']}},
}
state = 'q0'
index = 0
sym = ''
while state != 'qT':

```



```
    sym, step, state = task[state][mem[index]]
    mem[index] = sym
    index += step
print(''.join(mem))
```