

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

Кривенцова Л.С.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Освоить основные принципы работы Машины Тьюринга.

Задание.

На вход программе подается строка неизвестной длины. Каждый элемент является значением в ячейке памяти ленты Машины Тьюринга.

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

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. 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 (MT) была разработана британским математиком, Аланом Тьюрингом (23.06.1912-07.06.1954), в 1936 году для изучения свойств алгоритмов и их вычисления на реальных устройствах. В дополнение к данной модели А. Тьюринг также ввел понятие вычислимости – свойства, определяющего возможность вычисления функции с помощью машины Тьюринга. Если для какой-то функции существует вычисляющая ее машина Тьюринга, то такую функцию называют алгоритмически вычислимой. При этом, далеко не каждая функция является вычислимой – решения многих математических проблем не входят в класс вычислимых функций, например проблема останова. 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. Всевозможные символы, которые могут храниться на ленте, образуют алфавит.

Важно отметить, что при рассмотрении машины Тьюринга принимается допущение о том, что алфавит всегда конечен.

Сточки зрения стороннего наблюдателя, 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 Тьюринга останавливает свою работу).

Выполнение работы.

Программа получает на вход строку данных, преобразуя её в список (функция `list()`). Переменным q и `index` присваивается начальное состояние и начальный индекс (равный нулю, т. к. по условию, автомат находится где-то слева от числа). Далее создаётся словарь со словарями, моделирующий таблицу состояний. В каждом внутреннем словаре содержится список, с тремя элементами:

- 1) Символ, который записывается в текущую клетку
- 2) Направление движения автомата
- 3) Следующее состояние

Далее начинается `while`, моделирующий движение автомата по ленте Машины Тьюринга. Цикл завершается, когда состояние становится конечным (q_T). Затем производится вывод полученной ленты, при помощи функции `print()` и метода `join()`.

Таблица состояний:

	+	-	0	1	2	' '
q0	'+',R,q1	'-',R,q4	'0',R,q0	'1',R,q0	'2',R,q0	' ',R,q0
q1	'+',N,qT	'-',N,qT	'0',N,qT	'1',L,q2	'2',L,q3	' ',N,qT
q2	'+',L,q2	'-',L,q2	'1',N,qT	'2',N,qT	'0',L,q2	'1',N,qT
q3	'+',L,q3	'-',L,q3	'2',N,qT	'0',L,q2	'1',L,q2	'1',N,qT
q4	'+',N,qT	'-',N,qT	'0',N,qT	'1',L,q5	'2',L,q6	' ',N,qT
q5	'+',L,q5	'-',L,q5	'2',L,q5	'0',R,q7	'1',N,qT	' ',N,qT
q6	'+',L,q6	'-',L,q6	'1',L,q5	'2',L,q5	'0',R,q7	' ',N,qT
q7	'+',N,qT	'-',N,qT	'0',L,q8	'1',L,q7	'2',L,q7	' ',N,qT
q8	'+',N,qT	'-',N,qT	'0',N,qT	'1',N,qT	'2',N,qT	' ',R,q9
q9	'+',N,qT	'-',N,qT	' ',N,qT	'1',N,qT	'2',N,qT	' ',R,q9

q0 - начальное состояние, которое необходимо, чтобы найти знак «+» или «-».

q1 - состояние, которое необходимо, чтобы найти цифру справа от знака, которую нужно будет прибавить к первому числу.

q2 - состояние, которое необходимо, чтобы найти правую (последнюю) цифру первого числа, и прибавить к ней 1.

q3 - состояние, которое необходимо, чтобы найти правую (последнюю) цифру первого числа, и прибавить к ней 2.

q4 - состояние, которое необходимо, чтобы найти цифру справа от знака, которую нужно будет отнять от первого числа.

q5 - состояние, которое необходимо, чтобы найти правую (последнюю) цифру первого числа, и отнять от неё 1.

q6 - состояние, которое необходимо, чтобы найти правую (последнюю) цифру первого числа, и отнять от неё 2.

q7 - состояние, которое необходимо, чтобы проверить, что находится справа от только что записанного нуля.

q8 - состояние, которое необходимо, чтобы проверить, что находится слева от только что записанного нуля.

q9 - состояние, которое необходимо, чтобы заменить на пробел незначащий ноль.

Переменные:

memory — список, текущее состояние ленты;

index — переменная, содержащая целое число — индекс клетки, где на ленте Машины Тьюринга находится автомат;

q — строковая переменная, содержащая текущее состояние Машины Тьюринга;

table — словарь со словарями, моделирующий таблицу состояний;

sum - переменная, содержащая целое число — индекс клетки, куда на ленте Машины Тьюринга следует перейти автомату;

delta - переменная, содержащая целое число — сдвиг по клеткам ленты Машины Тьюринга;

state - строковая переменная, содержащая состояние Машины Тьюринга, в которое следует перейти.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	111+1	112+1	Программа работает верно.
2.	100-1	22-1	Программа работает верно.
3.	221+2	1000+2	Программа работает верно.
4.	111-2	102-2	Программа работает верно.

Выводы.

Были изучены принципы работы Машины Тьюринга.

Разработана программа, моделирующая действие Машины Тьюринга, принимающая на вход строку(ленту), выполняющая операцию сложения(вычитания), и печатающая изменённую ленту.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2.py

```
memory = list(input())
index = 0
q = 'q0'
table = { 'q0' : { '+': ['+', 1, 'q1'], '-': ['- ', 1, 'q4'],
                    '0': ['0', 1, 'q0'], '1': ['1', 1, 'q0'],
                    '2': ['2', 1, 'q0'], ' ': [' ', 1, 'q0']}},
          'q1' : { '+': ['+', 0, 'qT'], '-': ['- ', 0, 'qT'],
                    '0': ['0', 0, 'qT'], '1': ['1', -1, 'q2'],
                    '2': ['2', -1, 'q3'], ' ': [' ', 0, 'qT']}},
          'q2' : { '+': ['+', -1, 'q2'], '-': ['- ', -1, 'q2'],
                    '0': ['1', 0, 'qT'], '1': ['2', 0, 'qT'],
                    '2': ['0', -1, 'q2'], ' ': ['1', 0, 'qT']}},
          'q3' : { '+': ['+', -1, 'q3'], '-': ['- ', -1, 'q3'],
                    '0': ['2', 0, 'qT'], '1': ['0', -1, 'q2'],
                    '2': ['1', -1, 'q2'], ' ': [' ', 0, 'qT']}},
          'q4' : { '+': ['+', 0, 'qT'], '-': ['- ', 0, 'qT'],
                    '0': ['0', 0, 'qT'], '1': ['1', -1, 'q5'],
                    '2': ['2', -1, 'q6'], ' ': [' ', 0, 'qT']}},
          'q5' : { '+': ['+', -1, 'q5'], '-': ['- ', -1, 'q5'],
                    '0': ['2', -1, 'q5'], '1': ['0', 1, 'q7'],
                    '2': ['1', 0, 'qT'], ' ': [' ', 0, 'qT']}},
          'q6' : { '+': ['+', -1, 'q6'], '-': ['- ', -1, 'q6'],
                    '0': ['1', -1, 'q5'], '1': ['2', -1, 'q5'],
                    '2': ['0', 1, 'q7'], ' ': [' ', 0, 'qT']}},
          'q7' : { '+': ['+', 0, 'qT'], '-': ['- ', 0, 'qT'],
                    '0': ['0', -1, 'q8'], '1': ['1', -1, 'q7'],
                    '2': ['2', -1, 'q7'], ' ': [' ', 0, 'qT']}},
          'q8' : { '+': ['+', 0, 'qT'], '-': ['- ', 0, 'qT'],
                    '0': ['0', 0, 'qT'], '1': ['1', 0, 'qT'],
                    '2': ['2', 0, 'qT'], ' ': [' ', 1, 'q9']}},
          'q9' : { '+': ['+', 0, 'qT'], '-': ['- ', 0, 'qT'],
                    '0': [' ', 0, 'qT'], '1': ['1', 0, 'qT'],
                    '2': ['2', 0, 'qT'], ' ': [' ', 0, 'qT']}}

}

while q != 'qT':
    sym, delta, state = table[q][memory[index]]
    memory[index] = sym
    index += delta
    q = state

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