МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Математического обеспечения электронно-вычислительных машин

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Информатика»

ТЕМА: МОДЕЛИРОВАНИЕ РАБОТЫ МАШИНЫ ТЬЮРИНГА

Студентка гр. 0382 _____ Рубежова Н.А.

Преподаватель _____ Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Изучить основные принципы работы Машины Тьюринга, а также отработать ее реализацию и моделирование на языке Python.

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

1. 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.

2. 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.

3. 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: q0, q1, q2 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.

Выполнение работы.

1. Составим таблицу состояний Машины Тьюринга.

q1 – начальное состояние, поиск первой цифры первого числа.

q2 – поиск знака арифметической операции, заданной пользователем

q3 –поиск троичной цифры для сложения (на сколько будем увеличивать число)

q4 –выполнение операции «+1»

q5 –выполнение операции «+2»

q6 –состояние, которое необходимо для переноса разряда(единицы) при сложении

q7 –поиск троичной цифры для вычитания (на сколько будем уменьшать число)

q8 – выполнение операции «-1»

q9 – выполнение операции «-2»

q10 –состояние, которое необходимо, если мы заняли разряд при вычитании

q11 – проверка на то, значащий ли нуль мы поставили при «займе» из единицы

q12 –если поставленный в q10 нуль – незначащий (выяснили в q11), то «ликвидируем» его

	‘0’	‘1’	‘2’	‘+’	‘-’	‘ ’
q1 поиск первой цифры	-	‘1’,R,q2 #нашли первую цифру числа	‘2’,R,q2 #нашли первую цифру числа	-	-	‘ ’,R,q1 #прокручиваем пробелы
q2 поиск знака операции	‘0’,R,q2 #прокрутка числа	‘1’,R,q2 #прокрутка числа	‘2’,R,q2 #прокрутка числа	‘+’,R,q3 #нашли плюс, переходим к поиску цифры для сложения	‘-’,R,q7 #нашли минус, переходим к поиску цифры для вычитания	-
q3 на сколько будем увеличивать число? +разворот	‘0’,N,qT #пришли к операции «+0», не меняет число	‘1’,L,q4 #пришли к операции «+1», переходим к выполнению	‘2’,L,q5 #пришли к операции «+2», переходим к выполнению	-	-	-
q4 прокрутка знака+ операция «+1»	‘1’,N,qT #0+1=1 сложение завершено	‘2’,N,qT #1+1=2 сложение завершено	‘0’,L,q6 #2+1=0(1 в уме), должны перенести разряд	‘+’,L,q4 #прокрутка знака, чтобы перейти к самому числу	-	-
q5 прокрутка знака+ операция «+2»	‘2’,N,qT #0+2=2 сложение завершено	‘0’,L,q6 #1+2=0(1 в уме), нужен перенос	‘1’,L,q6 #2+2=1(1 в уме), нужен перенос	‘+’,L,q5 #прокрутка знака, чтобы перейти к самому числу	-	-
q6 перенос разряда при сложении (если нужно)	‘1’,N,qT #был нуль + 1 в уме =1	‘2’,N,qT #была 1 + 1 в уме =2	‘0’,L,q6 #было 2 + 1 в уме=0(1 в уме, повторим перенос)	-	-	‘1’,N,qT #разряды числа закончились, но у нас 1 в уме, поэтому добавляем разряд 1

q7 на сколько будем уменьшать число +разворот	‘0’,N,qT #пришли к операции «-0», не меняет число	‘1’,L,q8 #пришли к операции «-1», переходим к выполнению	‘2’,L,q9 #пришли к операции «-2», переходим к выполнению	-	-	-
q8 прокрутка знака +выполнение операции «-1»	‘2’,L,q10 #0-1, занимаем разряд, 3-1=2, так как заняли, переходим к след разряду	‘0’,N,qT #1-1=0 вычитание завершено	‘1’,N,qT #2-1=1 вычитание завершено	-	‘-‘,L,q8 #прокрутка знака, чтобы перейти к самому числу	-
q9 прокрутка знака +выполнение операции «-2»	‘1’,L,q10 #0-2 занимаем разряд, 3-2=1, так как заняли переходим к след разряду	‘2’,L,q10 #1-2 занимаем разряд, (1+3)-2=2, так как заняли переходим к след разряду	‘0’,N,qT #2-2=0 вычитание завершено	-	‘-‘,L,q9 #прокрутка знака, чтобы перейти к самому числу	-
q10 если при вычитании заняли один разряд	‘2’,L,q10 #из нуля занять не можем, занимаем у следующего	‘0’,L,q11 #заняли, значит, 1=>0, проверим, есть ли след. разряды, чтобы нуль был значащим	‘1’,N,qT #заняли, значит, 2=>1	-	-	-
q11 проверка значащий ли нуль мы поставили	‘0’,N,qT #есть значащий нуль, значит, перед ним точно есть цифра	‘1’,N,qT #есть единичка, значит, поставленный нуль значащий	‘2’,N,qT #есть двойка, значит, поставленный нуль значащий	-	-	‘ ’,R,q12 #разряд отсутствует, значит, поставленный нами нуль – незначащий, вернемся к нему

q12 ликвидация незначащего нуля	‘ ’,N,qT #убираем поставленный нами незначащий нуль	-	-	-	-	-

1. Опишем алгоритм в целом:

В состоянии q1 прокручиваем пробелы и ищем первую цифру первого числа. Переходим в q2.

В состоянии q2, прокрутив цифры первого числа, ищем знак запрашиваемой арифметической операции. Если это «+», переходим в q3 - поиск цифры для сложения, если « - » - переходим в q7 – поиск цифры для вычитания.

В состояниях q3-q6 работаем исключительно со сложением(условно, «блок сложения»), в q7-q12 – с вычитанием(условно, «блок вычитания»).

В состоянии q3 – ищем цифру, с которой будем складывать число: если «0» - то операция «+0» не меняет число, поэтому можем перейти в конечное(терминальное) состояние qT;
если «1» - переходим в состояние q4 – выполнение операции «+1»; если «2» - переходим в состояние q5 – выполнение операции «+2».

В состоянии q4 – выполняем с числом арифметическую операцию «+1». Машина Тьюринга считывает первый разряд числа(справа) и в соответствии с таблицей либо переходит в конечное состояние(если считываемый разряд «0»/«1»), либо переносит «единичку в уме» на следующий разряд(например, 2+1=>записываем нуль на месте двойки, «единичка в уме»), тогда МТ переходит в q6 и работает со следующим разрядом.

В состоянии q5 - выполняем с числом арифметическую операцию «+2». Машина Тьюринга считывает первый разряд числа(справа) и в соответствии с таблицей либо переходит в конечное состояние(если считываемый разряд «0»), либо переносит «единичку в уме» на следующий разряд(при «1»/«2»:

1+2=0(1 в уме), 2+2=1(1 в уме)), тогда МТ переходит в q6 и работает со следующим разрядом.

В состоянии q6 – складываем перенесенную «единичку в уме» с разрядом. Если считываемый разряд «0», МТ перезаписывает его на «1»(0+1=1) и переходит в qT, т.к. сложение завершено. Аналогично, «1» => «2». Если же считываемый разряд «2», то нам вновь понадобится перенос(2+1=0(1 в уме)), поэтому состояние не меняем, остаемся в q6 и повторяем операции.

В состоянии q7 – ищем цифру, которую будем вычитать из нашего числа:
если считывается «0» - то операция «-0» не меняет число, значит, МТ может перейти в конечное состояние qT, не изменив ленту;
если считывается «1» - то переходим в состояние q8 – для выполнения операции «-1»; если считывается «2», то переходим в состояние q9 – для выполнения операции «-2».

В состоянии q8 выполняем вычитание единицы из числа.
МТ считывает первый разряд(справа) числа.
Если «0», то из нуля вычесть единицу мы не можем, поэтому занимаем разряд у следующего, тогда будет (0+3)-1=2. МТ перезаписывает «0» на «2» и переходит в q10, чтобы тот разряд, у которого заняли, уменьшился на 1.
Если «1», то 1-1=0, следовательно, МТ перезаписывает «1» на «0» и завершает вычитание, переходя в конечное состояние qT.
Если «2», то 2-1=1, следовательно, МТ перезаписывает «2» на «1» и завершает вычитание переходя в конечное состояние qT.

В состоянии q9 выполняем арифметическую операцию «-2». МТ считывает первый разряд(справа) числа. Если считывается «0», то из нуля два мы не можем вычесть, значит, занимаем у следующего разряда, тогда (0+3)-2=1. МТ перезаписывает «0» на «1» и переходит в q10, чтобы тот разряд, у которого заняли, уменьшился на 1.
Если считывается «1», то снова не хватает для вычета двойки, занимаем у

следующего разряда, тогда (1+3)-2=2. МТ перезаписывает «1» на «2» и переходит в q10, чтобы тот разряд, у которого заняли, уменьшился на 1.

В состоянии q10 уменьшаем разряды, у которых «занимали» в процессе вычитания. МТ считывает разряд, у которого «занимали»: если считывается «0»(а он значащий), значит, мы снова должны «занять» у следующего разряда, то есть перейти в q10 и повторить операции; если считывается «1» , то так как мы «заняли» единичку, она уменьшается на 1 и МТ перезаписывает «1» на «0», но важно проверить, будет ли нуль, который мы поставили значащим, то есть есть ли за ним следующие разряды, поэтому МТ переходит в q11 – проверка значащий нуль или нет(есть ли слева от него разряд);

если считывается «2», то МТ перезаписывает «2» на «1» и завершает вычитание, переходя в qT.

В состоянии q11 проверяем, есть ли слева от нашего нуля какие-либо значащие цифры(1, 2 или даже 0, так как если встретится нуль, значит, он точно значащий, а значит за ним точно есть еще цифры), тогда поставленный нами в q10 нуль будет значимым. Следовательно, мы можем завершать вычитание, переходя в qT; если же слева от нашего нуля стоит « »(пробел), значит, разрядов больше нет, а значит поставленный нами нуль будет стоять первым в числе(читая слева направо) – незначащим. Мы должны его убрать – для этого развернемся и перейдем в состояние q12.

В состояние q12 мы «приходим» только лишь за тем, чтобы убрать поставленный нами незначащий нуль. Поэтому ничего кроме нуля МТ встретить и не может, она считывает его и заменяет на « »(пробел). Теперь мы избавились от незначащего нуля, можно переходит в qT, вычитание завершено.

2. Моделирование на языке Python.

Для удобства моделировать работу Машины Тьюринга будем, используя словари. Поэтому для МТ направления {L, R, N} будем

конвертировать в числа {-1,1,0} соответственно, которые затем будут прибавляться к индексу, чтобы двигаться по ленте памяти.

Сама лента памяти будет реализована в виде списка *memory*, который мы получим из входной строки. Для этого «обернем» входную строку *memoty=list(input()*

Инициализируем переменную индекса *ind=0,* изменяя которую мы сможем «передвигаться» по нашей ленте памяти.

Текущее состояние будет хранить в себе переменная *q*. Присваиваем ей начальное состояние *‘q0’.*

Работая со словарями, нам будет удобно возвращать тройку для МТ, которая будет выглядеть так: *[‘sym’, delta, q’],* где *‘sym’* – символ, которая МТ должна записать на месте разряда, *delta* – направление, куда МТ должна передвинуться(смещение индекса), *q’* – в какое состояние МТ должна перейти.

Инициализируем «двухуровневый» словарь *dict*, где ключами первого уровня будут текущие состояния МТ(q1-q12), а значению каждого из них будет соответствовать «вложенный» словарь, ключами которого являются возможные считываемые символы, а их значениям будет соответствовать та самая тройка *[‘sym’, delta, q’],* соответствующую таблице состояний.

Чтобы вернуть эту тройку, мы будем обращаться к элементу *dict[q][memory[ind]]*. А для того, чтобы удобно работать с каждым элементом этой тройки, «распарсим» ее с помощью такой строчки кода: *sym,delta,q = dict[q][memory[ind]]*.

Будем получать эти тройки в цикле *while,* пока наше состояние не станет конечным:
while q!=’qT’:

Изменяя считываемый символ *memory[ind]* на полученный из тройки *‘sym’,* а индекс *ind+=delta.*

После того, как состояние перейдет в конечное, программа выйдет из цикла.

Остается только вывести полученную ленту(наш список *memory)* на экран строкой. Сделаем это с помощью функции *printf(‘ ’.join(memory)).*

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
	111+1	112+1	Верный результат, формат вывода соответствует указанному.
	1-1	0-1	Верный результат, формат вывода соответствует указанному.
	11-2	2-2	Верный выведенный результат. Незначащие нули отсутствуют – верно.

Выводы.

Были изучены основные принципы работы Машины Тьюринга, а также отработаны на практике ее реализация и моделирование на языке Python.

Разработана программа, на вход которой подается строка неизвестной длины, представляющая собой «модель» ленты памяти Машины Тьюринга. Программа обрабатывает ленту, выполняет арифметическую операцию, заданную пользователем, и выводит полученную ленту после завершения работы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
memory=list(input())
ind=0
q='q1'
dict = {'q1': {'0': ['0', 1, 'q2'],
                '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, 'q7']
                },

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

        'q4' : {'0': ['1', 0, 'qT'],
                '1': ['2', 0, 'qT'],
                '2': ['0', -1, 'q6'],
                '+': ['+', -1, 'q4']
                },

        'q5' : {'0': ['2', 0, 'qT'],
                '1': ['0', -1, 'q6'],
                '2': ['1', -1, 'q6'],
                '+': ['+', -1, 'q5']
                },

        'q6' : {'0': ['1', 0, 'qT'],
                '1': ['2', 0, 'qT'],
                '2': ['0', -1, 'q6'],
                ' ': ['1', 0, 'qT']
                },
        'q7' : {'0': ['0', 0, 'qT'],
                '1': ['1', -1, 'q8'],
                '2': ['2', -1, 'q9']
                },
        'q8' : {'0': ['2', -1, 'q10'],
                '1': ['0', 0, 'qT'],
                '2': ['1', 0, 'qT'],
                '-': ['-', -1, 'q8']
                },
        'q9' : {'0': ['1', -1, 'q10'],
                '1': ['2', -1, 'q10'],
                '2': ['0', 0, 'qT'],
                '-': ['-', -1, 'q9']
                },
```

```
        'q10' : {'0': ['2', -1, 'q10'],
                 '1': ['0', -1, 'q11'],
                 '2': ['1', 0, 'qT']
                 },
        'q11' : {'0': ['0', 0, 'qT'],
                 '1': ['1', 0, 'qT'],
                 '2': ['2', 0, 'qT'],
                 ' ': [' ', 1, 'q12']
                 },
        'q12' : {'0': [' ', 0, 'qT']}
         }
while q!='qT':
    sym,delta,q = dict[q][memory[ind]]
    memory[ind]=sym
    ind+=delta
print(''.join(memory))
```