

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

Чегодаева Е.А.

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			
--	--	---	---	---	---	---	--	--	--

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.

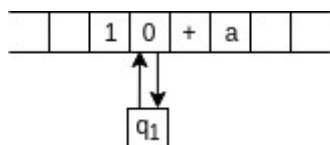
3. Числа и знак операции между ними идут непрерывно.

4. Гарантируется, что в результате операции вычитания не может получиться отрицательного числа.

Основные теоретические положения.

Машина Тьюринга

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.

Встроенные функции: *input()* , *print()* (Ввод и вывод соответственно).

Цикл: *While* <...> (Выполняет тело цикла, пока условие верно).

Метод: “...”*.join()* (Соединяет объекты символом <...>).

Выполнение работы.

На вход программы подаётся “Изначальное состояние ленты МТ” при помощи встроенной функции *input()* и преобразуется в список. Лента хранится в *memory*. Далее переменной *i*, олицетворяющей каретку, присваиваем 0 (т.к. каретка изначально слева). Следом идёт словарь *table*, представляющий таблицу состояний МТ (Таблицу см. в таблице 1). Затем, посредством цикла *while* реализована работа МТ: *sym* - запись нового символа, *delta* - шаг(вправо или влево), *state* - текущее состояние. Вывод “изменённой ленты” происходит посредством *print()* и метода “...”*.join()* (Для верного формата вывода)

Таблица 1.

	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10
0	0,1,q1	0,0,qT	1,0,qT	2,0, qT	0,0, qT	2,-1,q6	1,-1,q6	0,0,qT	0,0,qT	','1, q10
1	1,1,q1	1,-1,q3	2,0,qT	0,-1,q3	1,-1,q6	0,-1,q9	2,-1,q6		1,0,qT	1,0,qT
2	2,1,q1	2,-1,q4	0,-1,q3	1,-1,q3	2,-1,q7	1,0,qT	0,0,qT		2,0,qT	2,0,qT
+	+,1,q2		+, -1,q3	+, -1,q4						
-	-,1,q5					-, -1,q6	-, -1,q7		-,0,qT	-, -1,q8
''	','1,q1		1,0,qT						','1,q10	

- q1 - начальное состояние, определяет арифметическую операцию (если встречается “+” → переход в состояние q2, “-” → состояние q5) , движется слева направо, символы не меняются.
- q2 - состояние для определения второго слагаемого (при '0' → программа переходит в qT, '1' → состояние q3, '2' → состояние q4), при любом символе делает шаг влево, символы не меняются.
- q3 - состояния, реализующие “+1”, движется справа налево до первой цифры первого слагаемого. При встрече '0' → '1' и переход в состояние qT; '1' → '2', qT; При встрече '2' необходимо перенести 1 в старший разряд, '2' → '0' , сделать один шаг влево и повтор действий этого состояния, но со старшими разрядами; ' ' → '1' и переход в qT.
- q4 - состояние, реализующие “+2”, движется справа налево до первой цифры первого слагаемого. При встрече '0' → '2' , переход в состояние qT; При встрече '1' → '0' и переход в состояние q3, так как необходимо добавить к старшему разряду 1; '2' → '1' , q3 (аналогично встрече '1').
- q5 - состояние для определения вычитаемого (при '0' → программа переходит в qT, '1' → состояние q6, '2' → состояние q7), при любом символе делает шаг влево, символы не меняются.
- q6 - состояние, реализующие “-1”, движется справа налево до первой цифры первого числа. При встрече '0' → '2', один шаг влево и повтор действий этого состояния, но со старшими разрядами; При встрече '1' → '0' и переход в q9; При встрече '2' → '1', qT.
- q7 - состояние, реализующие “-2”, движется справа налево до первой цифры первого числа. При встрече '0' → '1' , один шаг влево и переход в q6, так как необходимо отнять 1 от старших разрядов; При встрече '1' → '2' и действия, аналогичные встрече '0'; При встрече '2' → '0' и переход в состояние qT.
- q8,q9,q10 - состояния, реализующие удаление '0' из старших разрядов и проверку частных случаев: “1-1” и “2-2”.
- qT - состояние, означающее завершение работы МТ.

Тестирование.

Результаты тестирования представлены в табл. 2.

Таблица 2 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	1-1	0-1	Ответ верный.
2.	121+2	200+2	Ответ верный.
3.	100010-2	100001-2	Ответ верный.
4.	100-2	21-2	Ответ верный.
5.	111-1	110-1	Ответ верный.
6.	1001-2	222-2	Ответ верный.
7.	2-1	1-1	Ответ верный.
8.	21+2	100+2	Ответ верный.
9.	112+2	121+2	Ответ верный.
1 0.	2022+1	2100+1	Ответ верный.

Выводы.

Была изучена и смоделирована работа Машины Тьюринга.

Разработана программа, моделирующая работу Машины Тьюринга, выполняющая считывание “исходной ленты” с клавиатуры с помощью функции *input()*, обработку данных посредством словаря и цикла *while*. Ответ к поставленной задаче выводиться с помощью функции *print()* и метода *.join()*.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb2.py

```
memory=list(input())
q='q1'
i=0
table={ 'q1': { '0': ['0',1, 'q1'], '1':['1',1,'q1'], '2':['2',1,'q1'], '+':['+',1,'q2'],
'-': ['- ',1,'q5'], ' ': [' ',1,"q1"]},
'q2': { '0': ['0',0, 'qT'], '1':['1',-1,'q3'], '2':['2',-1,'q4']},
'q3': { '0': ['1',0, 'qT'], '1':['2',0,'qT'], '2':['0',-1,'q3'], '+':['+',-1,'q3'], ' ': ['1',0,"qT"]},
'q4': { '0': ['2',0, 'qT'], '1':['0',-1,'q3'], '2':['1',-1,'q3'], '+':['+',-1,'q4']},
'q5': { '0': ['0',0, 'qT'], '1':['1',-1,'q6'], '2':['2',-1,'q7']},
'q6': { '0': ['2',-1, 'q6'], '1':['0',-1,'q9'], '2':['1',0,'qT'], '-': ['- ',-1,'q6']},
'q7': { '0': ['1',-1, 'q6'], '1':['2',-1,'q6'], '2':['0',0,'qT'], '-': ['- ',-1,'q7']},
'q8': { ' ': ['0',0,"qT"]},
'q9': { '0': ['0',0, 'qT'], '1':['1',0,'qT'], '2':['2',0,'qT'], '-': ['- ',0,'qT'], ' ': [' ',1,"q10"]},
'q10': { '0': [' ',1, 'q10'], '1':['1',0,'qT'], '2':['2',0,'qT'], '-': ['- ',-1,'q8']}
}
while q!='qT':
    sym, delta ,state=table[q][memory[i]]
    memory[i]=sym
    i+=delta
    q=state
print("".join(memory))
```