

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.01– «Информатика и вычислительная техника»

Лабораторная работа № 1
по дисциплине
«Основы алгоритмизации и программирования»
на тему
«Машина Тьюринга»

Выполнил студент гр. ИВТ-23-16
Бучинский Ян Викторович

Проверил:

доцент кафедры ИТАС
Ярулин Денис Владимирович

(оценка)

(подпись)

(дата)

г. Пермь, 2023

Цели и задачи

Целью данной работы является создание алгоритма АМТ для сложения чисел из алфавита $A = \{ 1,2,3,4,5,6,7,8,9 \}$ в соответствии со структурой абстрактной машины Тьюринга.

Задачи:

- 1) Изучить структуру АМТ
- 2) Составить алгоритм АМТ
- 3) Оформить отчет

1 Разработка алгоритмов работы машины Тьюринга

1.1 Постановка задачи

Составить алгоритм для абстрактной машины Тьюринга (АМТ), который будет прибавлять к 9 к любому заданному числу составленному из алфавита.

Изначальное положение головы - конец числа.

1.2 Обход ограничений

Так как в данном алгоритме отсутствует символ «0», есть два вариант составления алгоритма с учетом данного ограничения:

- 1) Совершать действия в 9-ой системе счисления.
- 2) Заменить символ «0» на пустоту, которая есть в любом алфавите АМТ (исходя из структуры АМТ).

Однако во втором случае у нас возникает проблема различия «значащей пустоты»(которая в нашем случае заменяет «0») от «незначащей пустоты», если «значащая пустота» стоит в конце числа, например: $91 + 9 = 100$. Для этого есть два варианта разрешения данной ситуации:

- 1) Запоминать индекс ячейки начального положения головки АМТ.
- 2) Вводить вместе с числом символ, которого нет в алфавите АМТ.

Отталкиваясь от структуры АМТ, если построить программу, которая не будет заходить на данный символ, то ее работа продолжится.

Таким образом мы сможем обозначить для пользователя конец числа.

1.3 Словесный алгоритм

Q1 - начальное состояние АМТ, которое прибавляет 9 к числу в ячейке и заменяет его на младший разряд получившейся суммы, запоминает старший(т.е. 1, если это не ситуация $0+9$) и сдвигается влево, меняя состояние на Q2.

Пример: $8+9 = 17$, следовательно АМТ перезапишет символ «8» на «7» и сделает сдвиг влево.

Q2 - прибавляет к числу 1 и перезаписывает символ ячейки на 1 больше с последующей остановкой, кроме случая, если в ячейке был символ «9». Тогда происходит перезапись на «значащую пустоту» с последующим сдвигом и повторном запуском состояния Q2. Пример таблицы переходов на рисунке 1.

	Q ₁	Q ₂
1	<u> </u> ← Q ₂	2 ←
2	1 ← Q ₂	3 ←
3	2 ← Q ₂	4 ←
4	3 ← Q ₂	5 ←
5	4 ← Q ₂	6 ←
6	5 ← Q ₂	7 ←
7	6 ← Q ₂	8 ←
8	7 ← Q ₂	9 ←
9	8 ← Q ₂	<u> </u> ← Q ₂
	9 ←	1 ←

Рисунок 1

1.4 Разбор примера на ленте

Состояние							♦			
Q1			9	9	9	9	9			
							♦			
Q2			9	9	9	9	8			
							♦			
Q2			9	9	9	<u> </u>	8			
							♦			
Q2			9	9	9	<u> </u>	8			
							♦			
Q2			9	9	<u> </u>	<u> </u>	8			
							♦			
Q2			9	<u> </u>	<u> </u>	<u> </u>	8			
							♦			
Q2			<u> </u>	<u> </u>	<u> </u>	<u> </u>	8			
							♦			
Q2		1	<u> </u>	<u> </u>	<u> </u>	<u> </u>	8			

Было получено число 100008, проверим. $9999+9 = 100008$. Результат совпадает.

1.5 Скриншоты

Варианты входных и выходных данных представлены на рисунках 1 и 2, 3 и 4, 5 и 6 соответственно.

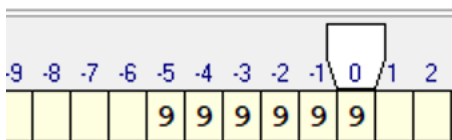


Рисунок 1 - Начальное состояние

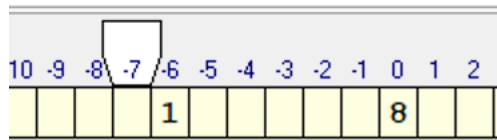


Рисунок 2 - Конечное состояние

Проверка: $99999+9 = 1000008$

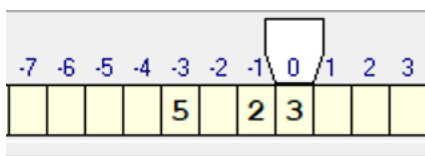


Рисунок 3 - Начальное состояние

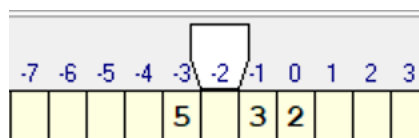


Рисунок 4 - Конечное состояние

Проверка: $5023+9 = 5032$

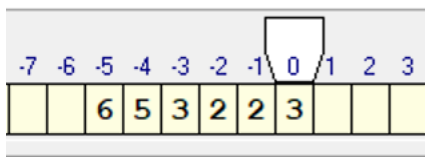


Рисунок 5 - Начальное состояние

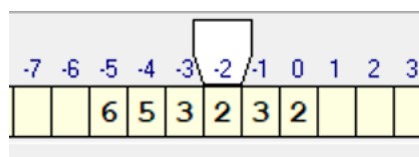


Рисунок 6 - Конечное состояние

Проверка: $653223+9 = 653232$

Работа с git на рисунках 7,8 и 9.

