

# ЛАБОРАТОРНАЯ РАБОТА №1

## ПЛАН

По дисциплине: Теория программирования  
Тема занятия: **Абстрактная машина Поста**  
Цель занятия: научиться реализовывать алгоритмы с помощью машины Поста  
Количество часов: 4

### Содержание работы:

1. Реализовать алгоритм умножения заданного числа на 2 с помощью машины Поста. В задаче учесть, что одна метка на ленте означает число 0.
2. Реализовать алгоритм с помощью машины Поста. Дано предложение, состоящее из слов, представленных разным количеством меток. Слова разделены одним пробелом. Добавить слово, состоящее из количества меток равного количеству пробелов между словами.

Пример: начальное предложение: vvv vv vvv

Результат выполнения алгоритма: vvv vv vvv vv

### Дополнительная информация

**Машина Поста** – это абстрактная вычислительная машина, созданная для уточнения (формализации) понятия алгоритма. Представляет собой универсальный исполнитель, позволяющий вводить начальные данные и читать результат выполнения программы.

В 1936 г. американский математик Эмиль Пост в статье описал систему, обладающую алгоритмической простотой и способную определять, является ли та или иная задача алгоритмически разрешимой. Если задача имеет алгоритмическое решение, то она представима в форме команд для машины Поста.

Машина Поста состоит из:

1. **бесконечной ленты**, поделенной на одинаковые ячейки (секции). Ячейка может быть пустой (0 или пустота) или содержать метку (1 или любой другой знак),
  2. **головки (каретки)**, способной передвигаться по ленте на одну ячейку в ту или иную сторону, а также способной проверять наличие метки, стирать и записывать метку.
- Текущее **состояние машины Поста** описывается состоянием ленты и положением каретки. **Состояние ленты** – информация о том, какие секции пусты, а какие отмечены. **Шаг** – это движение каретки на одну ячейку влево или вправо. Состояние ленты может изменяться в процессе выполнения программы.

Кареткой управляет программа, состоящая из строк команд. Каждая команда имеет следующий синтаксис:

**i K j**,

где **i** - номер команды, **K** – действие каретки, **j** - номер следующей команды (отсылка).

Всего для машины Поста существует шесть типов команд:

- **V j** - поставить метку, перейти к j-й строке программы.
- **X j** - стереть метку, перейти к j-й строке программы.
- **<- j** - сдвинуться влево, перейти к j-й строке программы.
- **-> j** - сдвинуться вправо, перейти к j-й строке программы.
- **? j<sub>1</sub>; j<sub>2</sub>** - если в ячейке нет метки, то перейти к j<sub>1</sub>-й строке программы, иначе перейти к j<sub>2</sub>-й строке программы.
- **!** – конец программы (стоп).

У команды «стоп» отсылки нет.

Варианты окончания выполнения программы на машине Поста:

1. **Команда "стоп"** - корректная остановка. Возникает в результате выполнения правильно написанного алгоритма.

2. **Выполнение недопустимой команды** – нерезультативная остановка. Случаи, когда головка должна записать метку там, где она уже есть, или стереть метку там, где ее нет, являются аварийными (недопустимыми).

3. **Уход в бесконечность, заикливание.** Машина Поста в результате работы алгоритма может вообще не остановиться (никогда не дойти до команды «стоп» и никогда не завершиться аварийной ситуацией).

Элементарные действия (команды) машина Поста проще команд машины Тьюринга. Поэтому программы для машины Поста имеют большее число команд, чем аналогичные программы для машины Тьюринга.

Почему достаточно лишь два различных символа (есть метка, нет метки)? Дело в том, что любой алфавит может быть закодирован двумя знаками; в зависимости от алфавита возрастет только количество двоичных символов в букве алфавита.

Пример работы машины Поста:

**Задача:** увеличить число 3 на единицу (изменить значение в памяти с 3 на 4).

Целое положительное число на ленте машины Поста представимо идущими подряд метками, которых на одну больше, чем кодируемое число. Это связано с тем, что одна метка обозначает ноль, а уже две – единицу, и т.д. Допустим, точно известно, что каретка стоит где-то слева от меток и обозревает пустую ячейку. Тогда программа увеличения числа на единицу может выглядеть так:

1 -> 2

2 ? 1,3

3 <- 4

4 V 5

5 !

А процесс выполнения может быть таким:

