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»

**Отсчёт № 2**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

**на тему «Решение задач для машины Тьюринга и алгоритмов Маркова»**

Выполнил студент гр. ИВТ-24-1б

Храмцов Илья Владимирович

Проверил:

(оценка) (подпись)

(дата)

г. Пермь, 2024

Содержание

[Введение 3](#_Toc180614169)

[Основная часть 4](#_Toc180614170)

[Машина Тьюринга. Задача 1. 4](#_Toc180614171)

[Машина Тьюринга. Задача 2 5](#_Toc180614172)

[Машина Тьюринга. Задача 3 6](#_Toc180614173)

[Заключение 7](#_Toc180614174)

[Список использованных источников 8](#_Toc180614175)

# Введение

Целью работы было реализовать решение шести задач - трёх для машины Тьюринга и трёх для алгоритмов Маркова.

Задачи:

1. Сформировать краткое словесное описание решения задач.
2. Реализовать решение задач в соответственных средах имитации машины Тьюринга либо алгоритмов Маркова

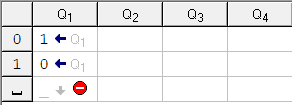
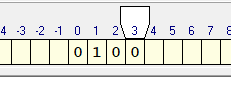
# Основная часть

## Машина Тьюринга. Задача 1.

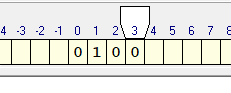
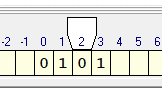
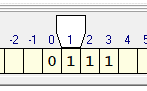
Постановка задачи: дано число, состоящее из 0 и 1. Заменить все 0 на 1 и 1 на 0. Головка находится на последней цифре числа.

Краткий словесный алгоритм: так как головка точно находится на конце числа, мы можем сразу считать значение ячейки под головкой и перейти к состоянию Q1 : если ячейка равняется 1, то мы заменяем её на 0, делаем сдвиг влево и остаёмся в исходном состоянии Q1 ; если ячейка равняется 0, то мы заменяем её на 1, делаем сдвиг влево и остаёмся в исходном состоянии Q1 ; если ячейка пуста, то головка уже прошла все заполненные ячейки и алгоритм следует завершить терминальным состоянием.

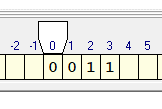
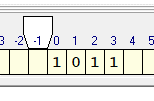
Визуализация:

Управляющее устройство Лента ячеек с примером числа

Шаг 0 Шаг 1 Шаг 2

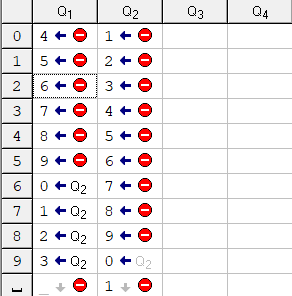
Шаг 3 Шаг 4

## Машина Тьюринга. Задача 2

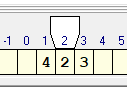
Постановка задачи: на ввод подаётся случайное число. Прибавить четыре. Головка находится на последней цифре числа.

Краткий словесный алгоритм: так как головка точно находится на конце числа, мы можем сразу считать значение ячейки под головкой и перейти к состоянию Q1 : если N + 4 меньше 10 (где N – цифра под головкой), то изменяем значение ячейки на N + 4 и завершаем алгоритм; если N + 4 >= 10, то изменяем значений ячейки на N + 4 – 10, головка делает сдвиг влево и алгоритм переходит в новое состояние Q2. В состоянии Q2 выполняется следующее: любая N кроме 9 изменяется на N + 1 и алгоритм принимает терминальное состояние; если N = 9 ячейка изменяется на 0, головка делает сдвиг влево и алгоритм остаётся в состоянии Q2; если ячейка под головкой пустая, то она заменяется на 1 и алгоритм принимает терминальное состояние.

Визуализация:

Управляющее устройство Лента ячеек с примером числа

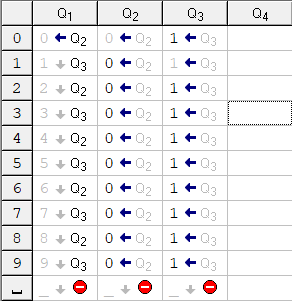
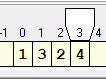
Шаг 0 Шаг 1 Шаг 2

## Машина Тьюринга. Задача 3

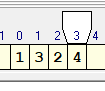
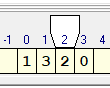
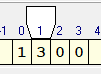
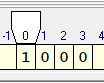
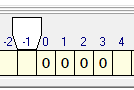
Постановка задачи: на ввод подаётся случайной число. Если число четное – заменить в нем все цифры на 0, иначе на 1. Головка находится на последней цифре числа.

Краткий словесный алгоритм: так как головка точно находится на конце числа, мы можем сразу считать значение ячейки под головкой и перейти к состоянию Q1: если N – чётное (где N – цифра под головкой), то и число является чётным, головка остаётся на месте и алгоритм переходит в новое состояние Q2 ; если N – нечётное, то и число является нечётным, головка остаётся на месте и алгоритм переходит в новое состояние Q3. В состоянии Q2 выполняется следующее: любая цифра заменяется на 0, головка делает сдвиг влево и остаётся в состоянии Q2; если под головкой пустая ячейка, она не изменяется и алгоритм принимает терминальное состояние. В состоянии Q3 выполняется следующее: любая цифра заменяется на 1, головка делает сдвиг влево и остаётся в состоянии Q3; если под головкой пустая ячейка, она не изменяется и алгоритм принимает терминальное состояние.

Визуализация:

Управляющее устройство Лента ячеек с примером числа

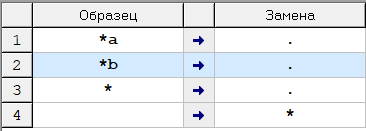
Шаг 0 Шаг 1 Шаг 2 Шаг 3 Шаг 4

## Алгоритмы Маркова. Задача 1

Постановка задачи: алфавит = {a, b}. Удалить из непустого слова P его первый символ. Пустое слово не менять.

Краткий словесный алгоритм: составим следующие правила подстановки : первым и вторым правилом будет терминальная замена подстроки \*a и \*b на пустую строку, мы с помощью четвёртого правила добавим перед строкой символ \*, а первое или второе правило удалит первый символ вместе с \* ; третьим правилом будет терминальная замена подстроки \* на пустую строку, так если бы у нас была пустая строка, то она бы не изменилась.

Визуализация:

Система подстановок Начальное состояние

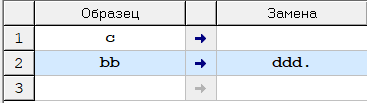
Шаг 0 Шаг 1 Шаг 2

## Алгоритмы Маркова. Задача 2

Постановка задачи: алфавит = {a, b, c, d}. В слове требуется удалить все вхождения c, а затем заменить первое вхождение bb на ddd.

Краткий словесный алгоритм: составим следующие правила подстановки: первым правилом будет нетерминальная замена подстроки с на пустую строку; вторым правилом будет терминальная замена подстроки bb на ddd

Визуализация:

Система подстановок Начальное состояние

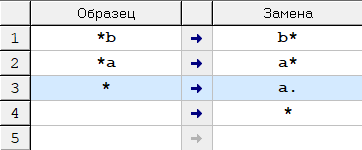
Шаг 0 Шаг 1 Шаг 2 Шаг 3

## Алгоритмы Маркова. Задача 3

Постановка задачи: алфавит = {a, b}, требуется приписать символ a к концу слова P.

Краткий словесный алгоритм: составим следующие правила подстановки: первым и вторым правилом будет нетерминальная замена подстроки \*a и \*b на a\* и b\* соответственно, мы добавим четвёртым правилом \* в начало слова, и правила 1 и 2 пронесут эту \* через все символы в конец слова ; третьим правилом будет терминальная замена \* на a, так мы добавим в конец слова подстроку a.

Визуализация:

Система подстановок Исходная строка

Шаг 0 Шаг 1 Шаг 2 Шаг 3 Шаг 4 Шаг 5

Шаг 6 Шаг 7

# Заключение

Из визуализации в средах имитации машины Тьюринга и алгоритмов Маркова видно, что алгоритм работает верно, и задачи были решены правильно.