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: 09.03.04 – «Программная инженерия»

**«Простые алгоритмы: Машина Тьюринга и алгоритм Маркова»**

Выполнил студент гр. РИС-24-3б

Гузий Степан Андреевич

Проверил:

Доцент кафедры ИТАС   
Ольга Андреевна Полякова

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (оценка) (подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

г. Пермь, 2024

**Машина Тьюринга (МТ)**

1. Постановка задачи №1

Дано: число, состоящие из Ø и 1. Алфавит = {a, b}

Цель: заменить все Ø на 1 и 1 на Ø

1. Алгоритм решения:

2.1 Головка стоит на первой цифре числа. Цифра считывается.

a) Если цифра Ø или 1, то головка двигается вправо пока не будет пустая строка. С помощью пустого символа определяем конец числа, после чего возвращаемся на последнюю цифру числа и переходим в состояние Q2.

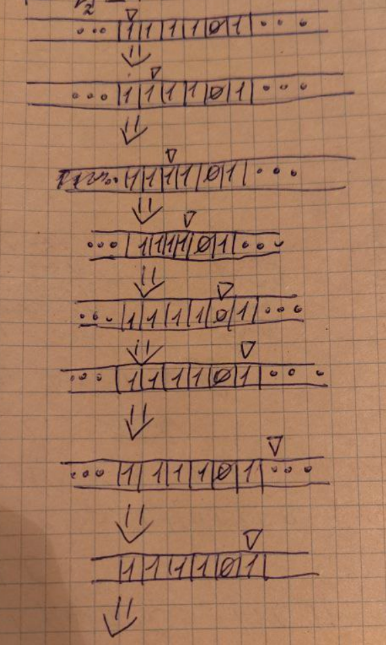
2.2 Головка стоит над последней цифре числа. Цифра под головкой считывается.

а) В состоянии Q2 заменяем все Ø на 1 и наоборот 1 на Ø. Параллельно двигаясь вправо. При триггере на пустой символ программа завершается.

3. Визуализация:

Таблица состояний:

|  |  |  |
| --- | --- | --- |
|  | Q1 | Q2 |
| Ø | Ø → Q1 | 1 ← Q2 |
| 1 | 1 → Q1 | Ø ← Q2 |
| «\_» | \_ ← Q2 | \_ ↓ 🛑 |



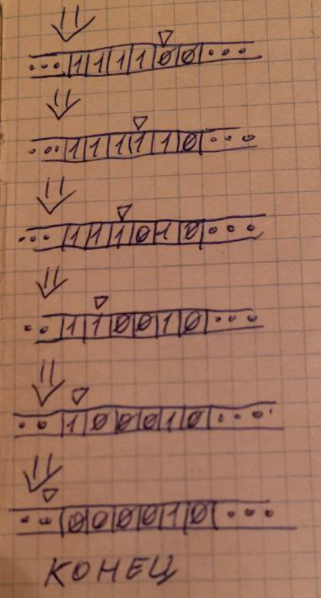


Рисунок 1 – работа программы

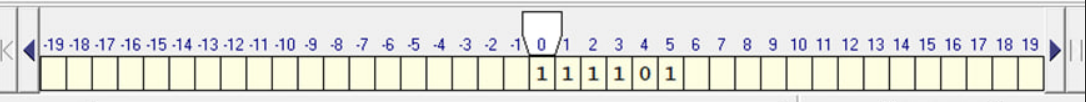


Рисунок 2 – визуализация ленты с головкой

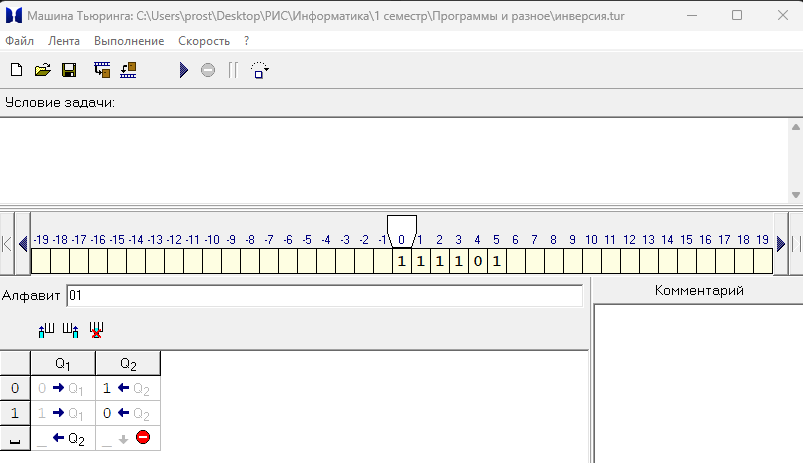
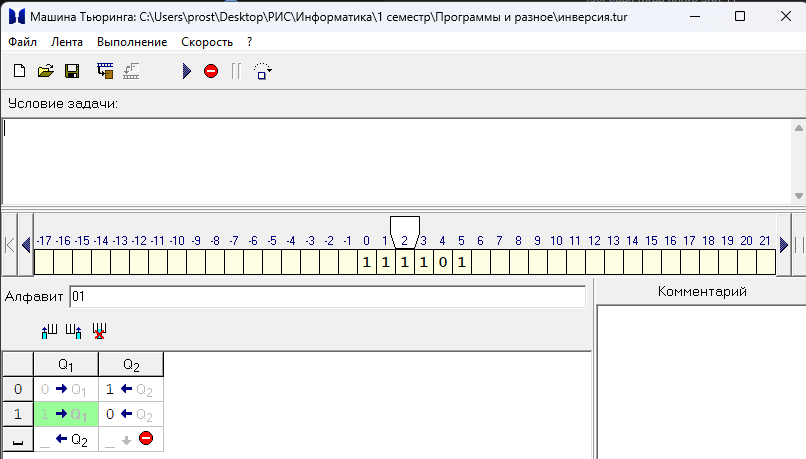
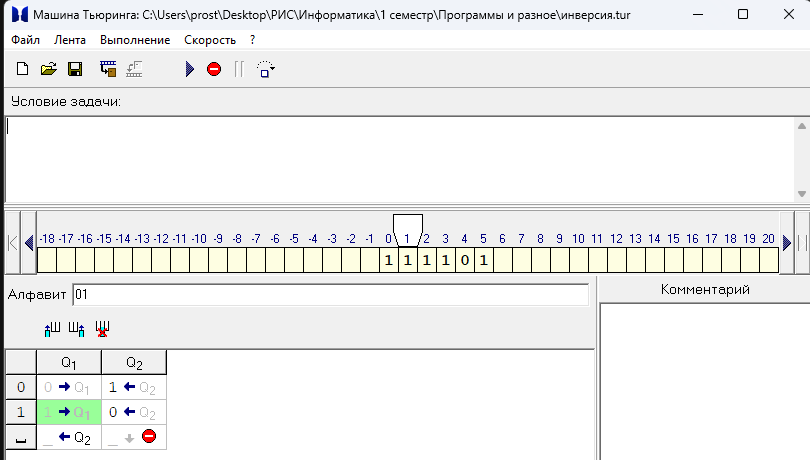
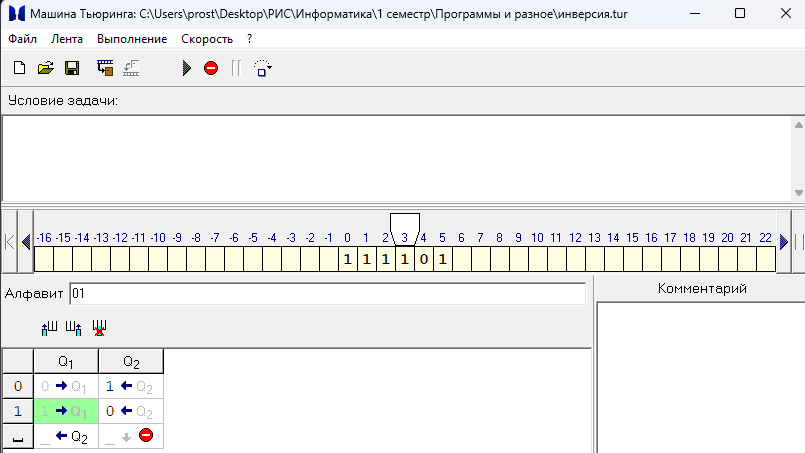
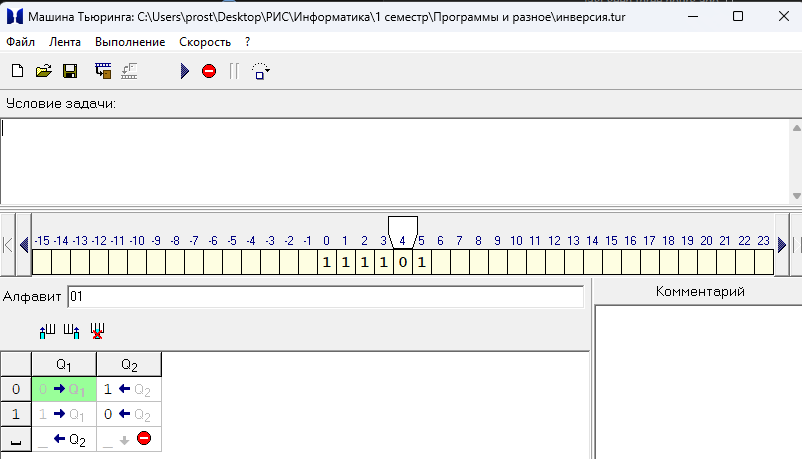


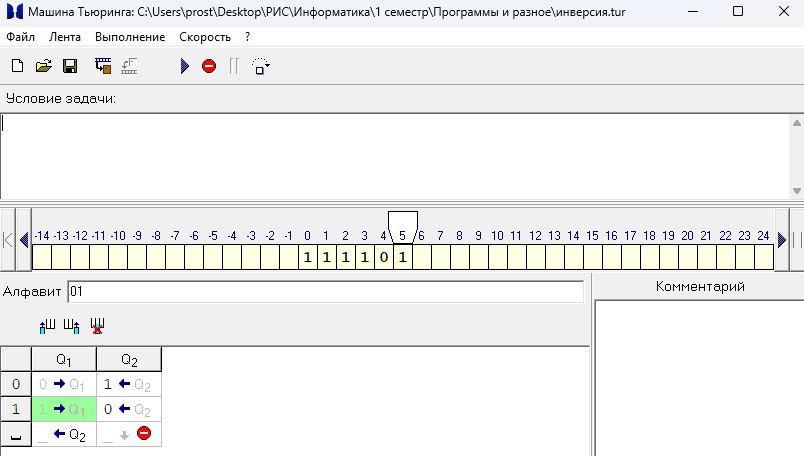
Рисунок 3 – тренажер МТ

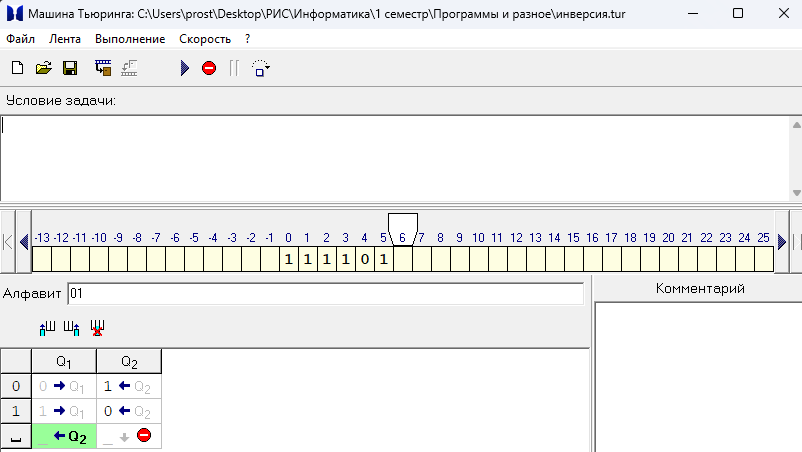












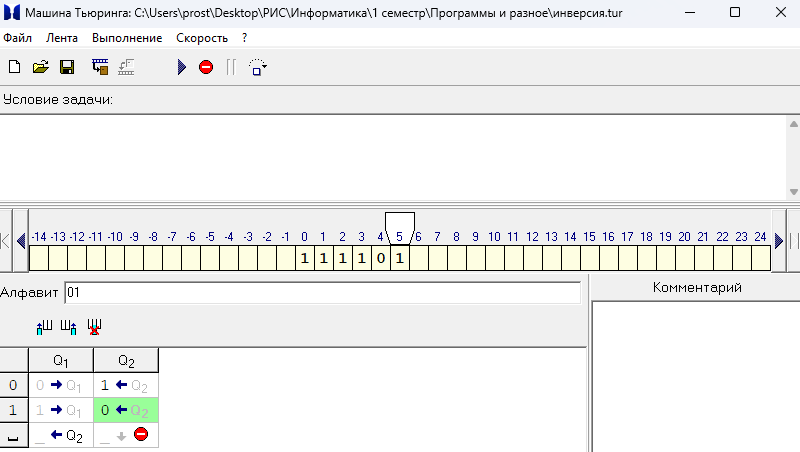
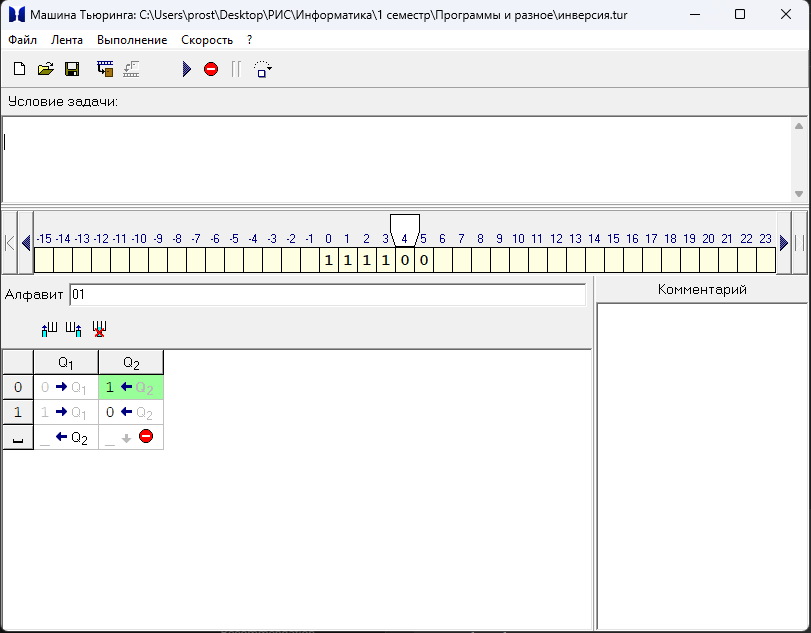
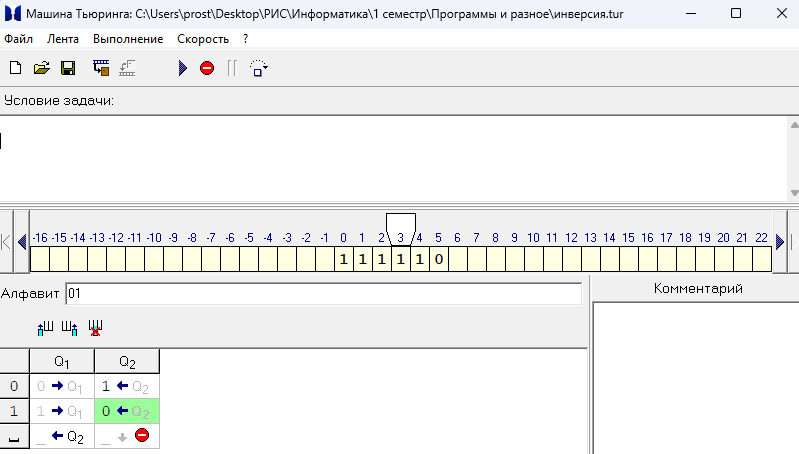
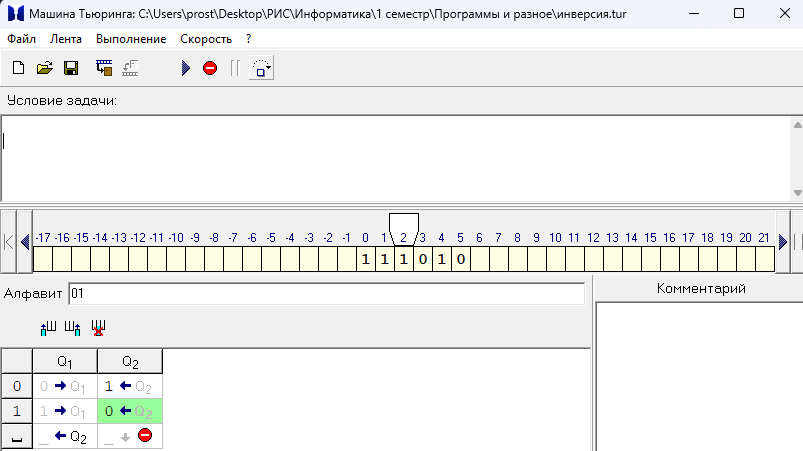
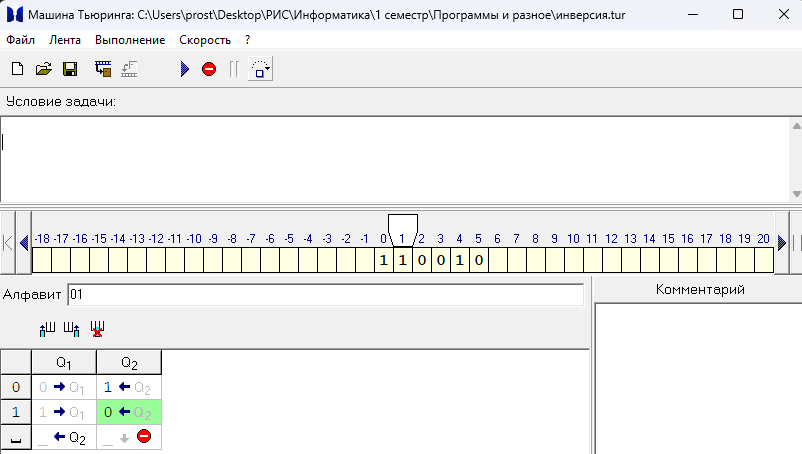


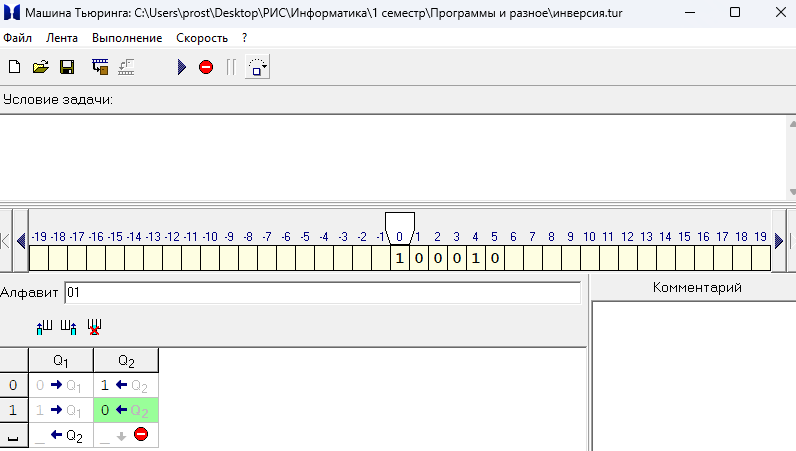
Рисунок 4 – смещение головки с помощью состояния Q1 и переход в состояние в Q2

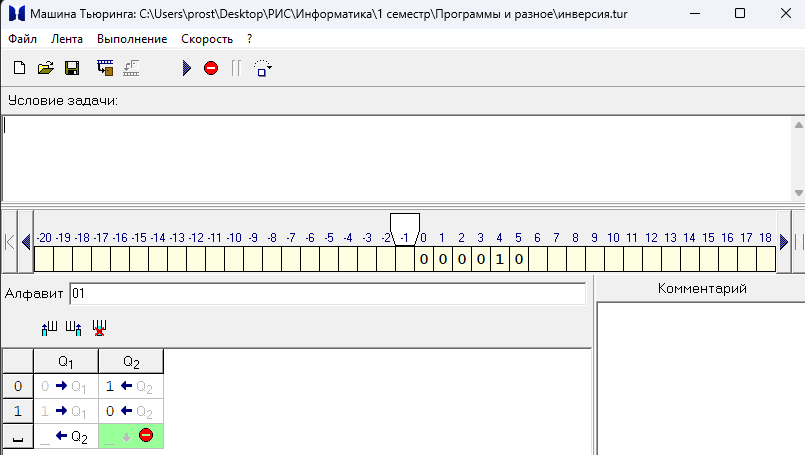












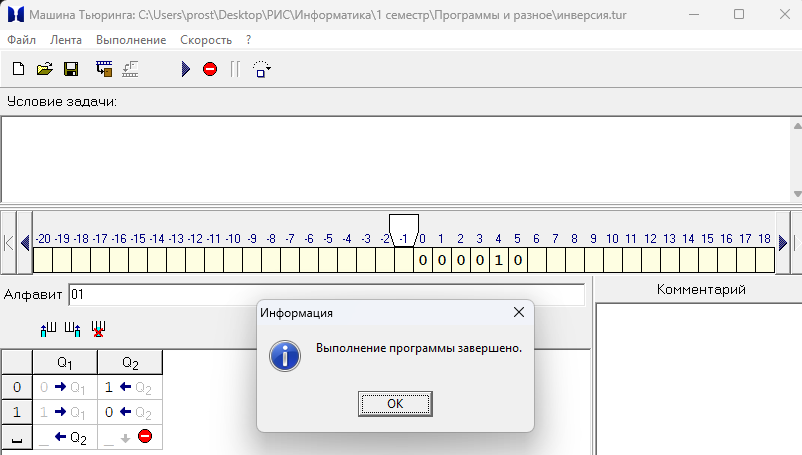


Рисунок 5 – замена чисел с помощью состояния Q2 и завершение программы

1. Постановка задачи №2

Дано: случайное число. Алфавит = {Ø,1,2,3,4,5,6,7,8,9}

Цель: прибавить четыре к случайному числу

1. Алгоритм решения:

2.1 Головка стоит на первой цифре числа. Цифра считывается.

a) Если цифра Ø-9, то головка двигается вправо пока не будет пустая строка. С помощью пустого символа определяем конец числа, после чего возвращаемся на последнюю цифру числа и переходим в состояние Q2.

2.2 Головка стоит над последней цифре числа. Цифра под головкой считывается

а) В состоянии Q2 прибавляем +4 к числам то если цифра Ø до 5, то Ø заменяется на 4, 1 на 5, 2 на 6, 3 на 7, 4 на 8, 5 на 9. Головка остается на месте и программа завершается.

б) Если идет переполнение разряда (например, 6+4=10) нужно сделать перенос единички в следующую ячейку слева на ленте. Если число 6 до 9, то 6 заменяется на 0, 7 на 1, 8 на 2, 9 на 3. Головка сдвигается влево, переходя в состояние Q3.

2.3 Если программа не завершилась. Цифра под головой считывается.

а) В состоянии Q3 мы добавляем по единичке в каждую ячейку от Ø до 8. Головка остается на месте, и программа завершается.

б) При значении числа 9, число меняется на Ø. Головка сдвигается влево. Программа переходит в состояние Q3.

2.4 Если над головкой находится пустая строка и программа не завершилась. Заменяем пустую строку на 1, и головка остается на месте. Программа завершается.

1. Визуализация:  
   Таблица состояний:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Q1 | Q2 | Q3 |
| Ø | Ø → Q1 | 4 ↓ 🛑 | 1 ↓ 🛑 |
| 1 | 1 → Q1 | 5 ↓ 🛑 | 2 ↓ 🛑 |
| 2 | 2 → Q1 | 6 ↓ 🛑 | 3 ↓ 🛑 |
| 3 | 3 → Q1 | 7 ↓ 🛑 | 4 ↓ 🛑 |
| 4 | 4 → Q1 | 8 ↓ 🛑 | 5 ↓ 🛑 |
| 5 | 5 → Q1 | 9 ↓ 🛑 | 6 ↓ 🛑 |
| 6 | 6 → Q1 | 0 ← Q3 | 7 ↓ 🛑 |
| 7 | 7 → Q1 | 1 ← Q3 | 8 ↓ 🛑 |
| 8 | 8 → Q1 | 2 ← Q3 | 9 ↓ 🛑 |
| 9 | 9 → Q1 | 3 ← Q3 | Ø ← Q3 |
| «\_» | \_ ← Q2 |  | 1 ↓ 🛑 |

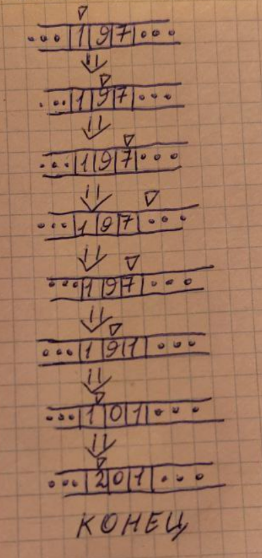


Рисунок 6 – работа программы

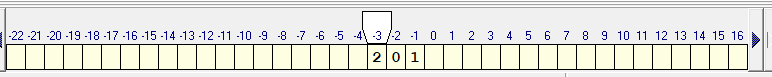


Рисунок 7 - визуализация ленты с головкой

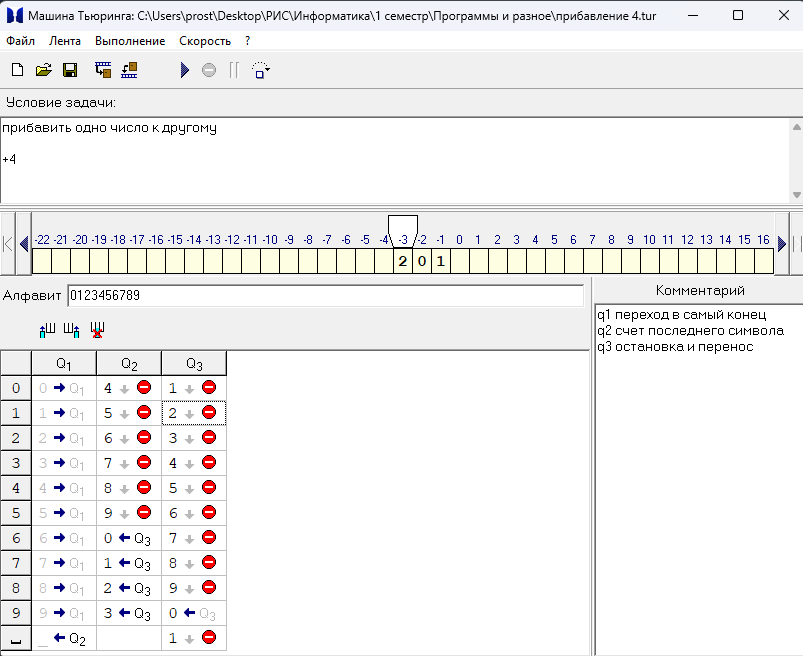
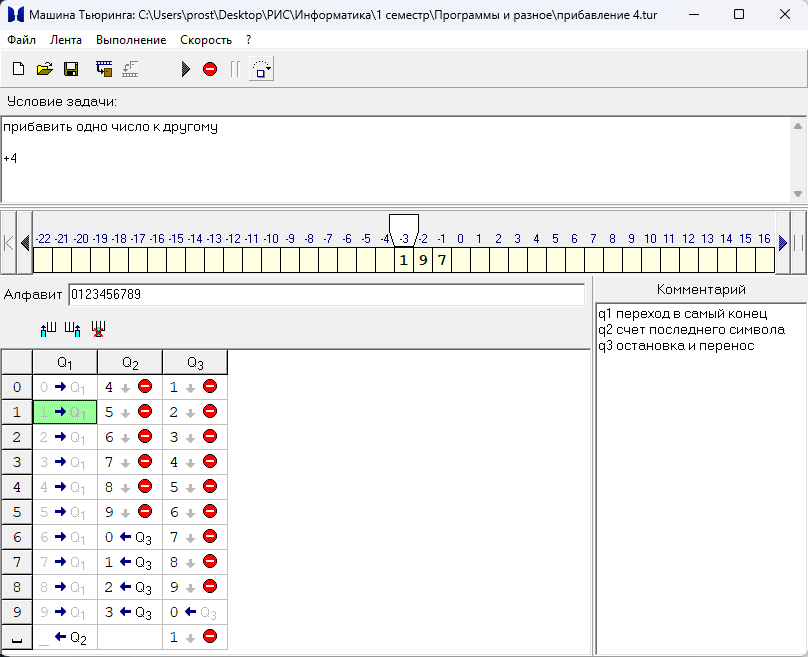
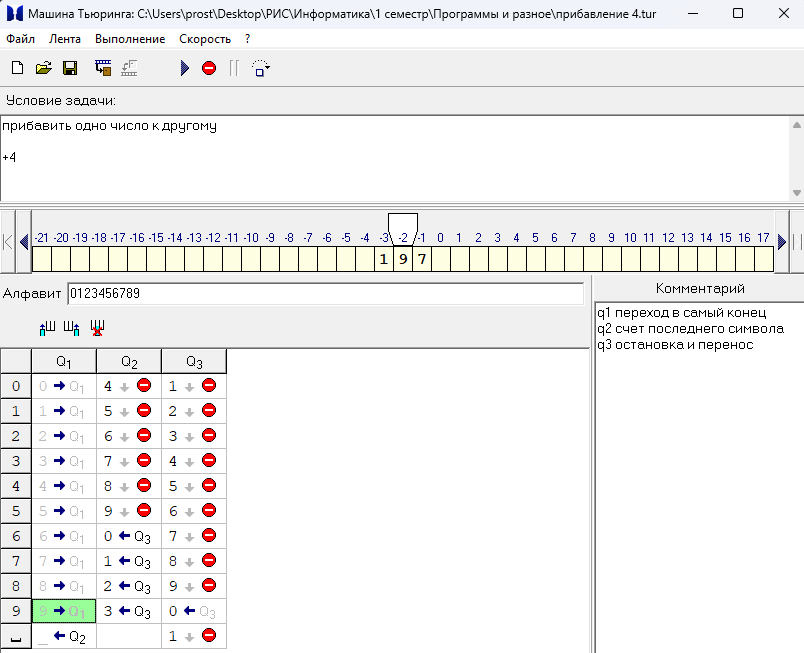
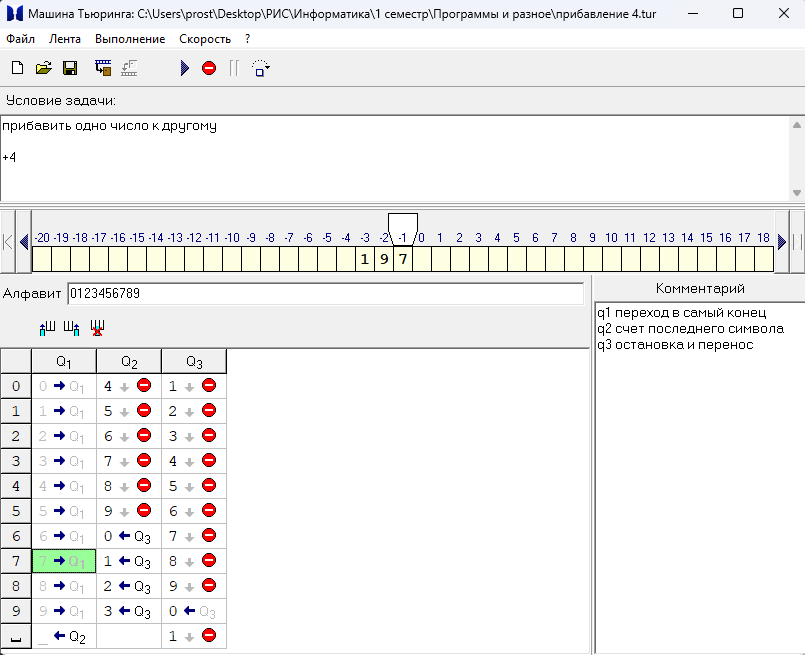
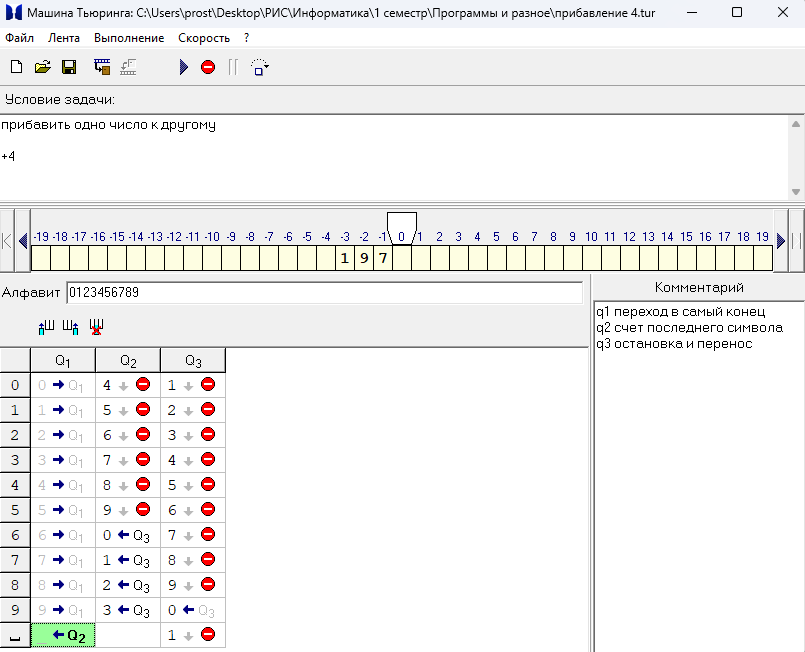


Рисунок 8 - тренажер МТ









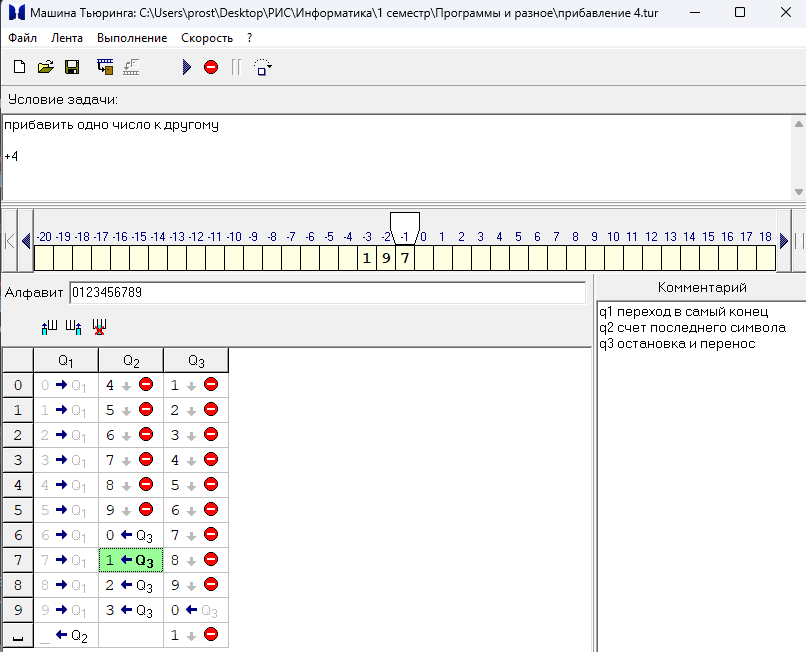


Рисунок 9 - смещение головки с помощью состояния Q1 и переход в состояние в Q2

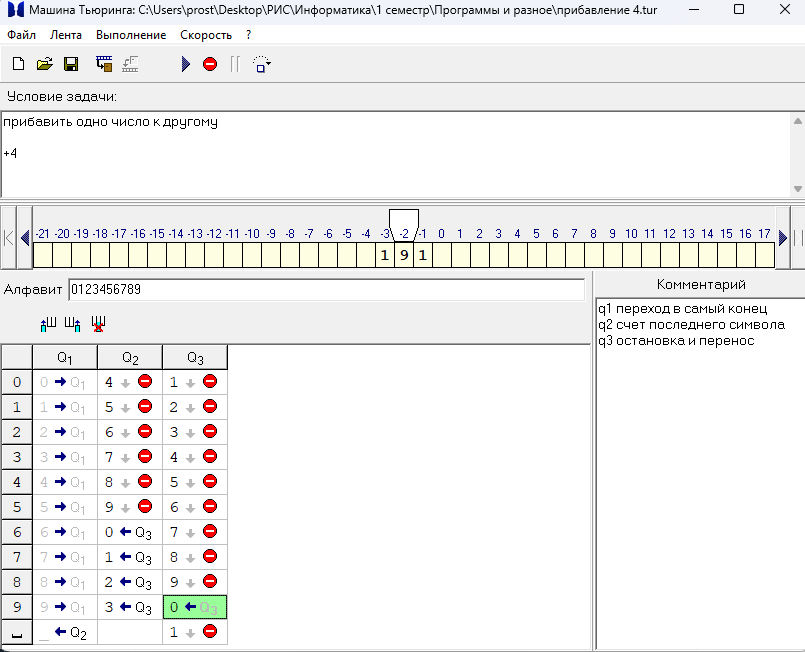
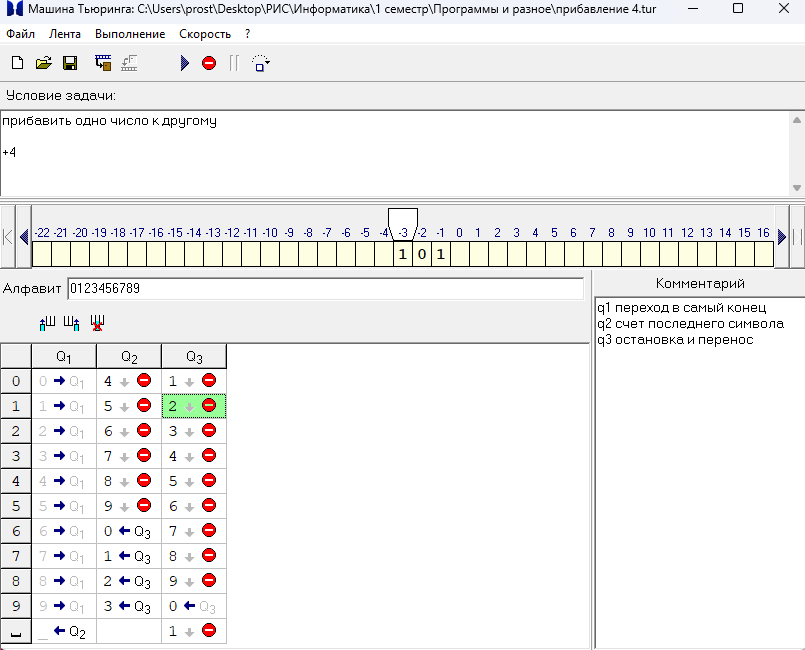


Рисунок 10 - замена символов с помощью состояния Q2 и переход в состояние Q3



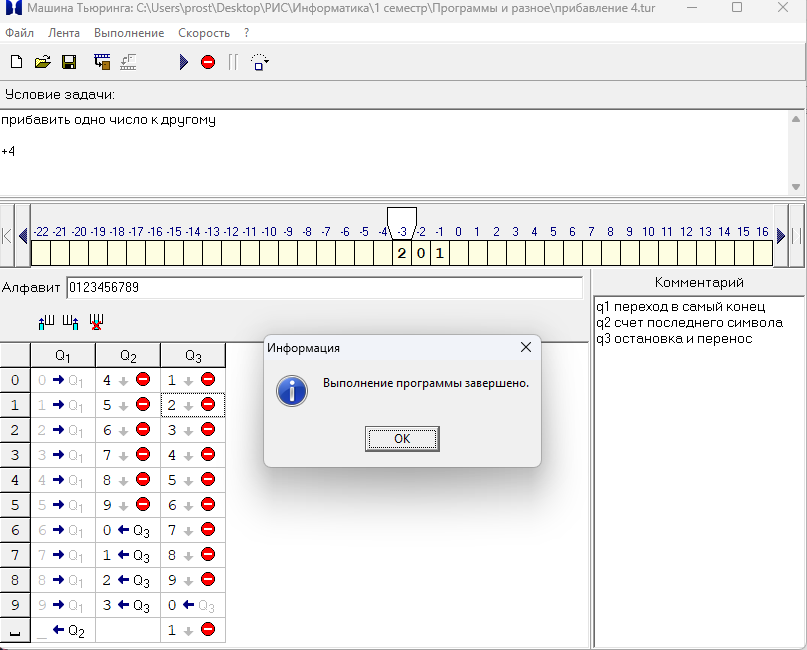


Рисунок 11 - замена чисел с помощью состояния Q3 и завершение программы

1. Постановка задачи №3

Дано: случайное число. Алфавит = {Ø,1,2,3,4,5,6,7,8,9}

Цель: если число четное заменить в нем все цифра на Ø, иначе на 1

1. Алгоритм решения:

2.1 Головка стоит на первой цифре числа. Цифра считывается.

a) Если цифра Ø-9, то головка двигается вправо пока не будет пустая строка. С помощью пустого символа определяем конец числа, после чего возвращаемся на последнюю цифру числа и переходим в состояние Q2.

2.2 Головка стоит над последней цифре числа. Цифра под головкой считывается

а) В состоянии Q2 алгоритм проверяет последнюю цифра числа. Она будет триггером что число четное (Ø, 2, 4, 6, 8) или нечетное (1, 3, 5, 7, 9). Если число четное, то головка остается на месте и переходит в состояние Q3.

б) Если число нечетное, то головка остается на месте, и программа переходит в состояние Q4.

2.3 Цифра под головой считывается. Происходит замена чисел.

а) В состоянии Q3 мы меняем число от Ø до 9 на Ø. Головка доходит до пустого символа и остается на месте. Программа завершается.

б) В состоянии Q4 мы меняем число от Ø до 9 на 1. Головка доходит до пустого символа и остается на месте. Программа завершается.

3. Визуализация:

Таблица состояний:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1 | Q2 | Q3 | Q4 |
| Ø | Ø → Q1 | Ø ↓ Q3 | Ø ← Q3 | 1 ← Q4 |
| 1 | 1 → Q1 | 1 ↓ Q4 | Ø ← Q3 | 1 ← Q4 |
| 2 | 2 → Q1 | 2 ↓ Q3 | Ø ← Q3 | 1 ← Q4 |
| 3 | 3 → Q1 | 3 ↓ Q4 | Ø ← Q3 | 1 ← Q4 |
| 4 | 4 → Q1 | 4 ↓ Q3 | Ø ← Q3 | 1 ← Q4 |
| 5 | 5 → Q1 | 5 ↓ Q4 | Ø ← Q3 | 1 ← Q4 |
| 6 | 6 → Q1 | 6 ↓ Q3 | Ø ← Q3 | 1 ← Q4 |
| 7 | 7 → Q1 | 7 ↓ Q4 | Ø ← Q3 | 1 ← Q4 |
| 8 | 8 → Q1 | 8 ↓ Q3 | Ø ← Q3 | 1 ← Q4 |
| 9 | 9 → Q1 | 9 ↓ Q4 | Ø ← Q3 | 1 ← Q4 |
| «\_» | \_ ← Q2 |  | \_ ↓ 🛑 | \_ ↓ 🛑 |

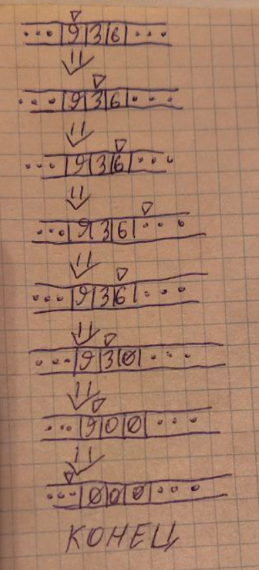


Рисунок 11 - работа программы

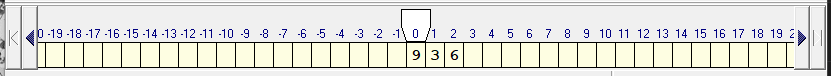


Рисунок 12 - визуализация ленты с головкой

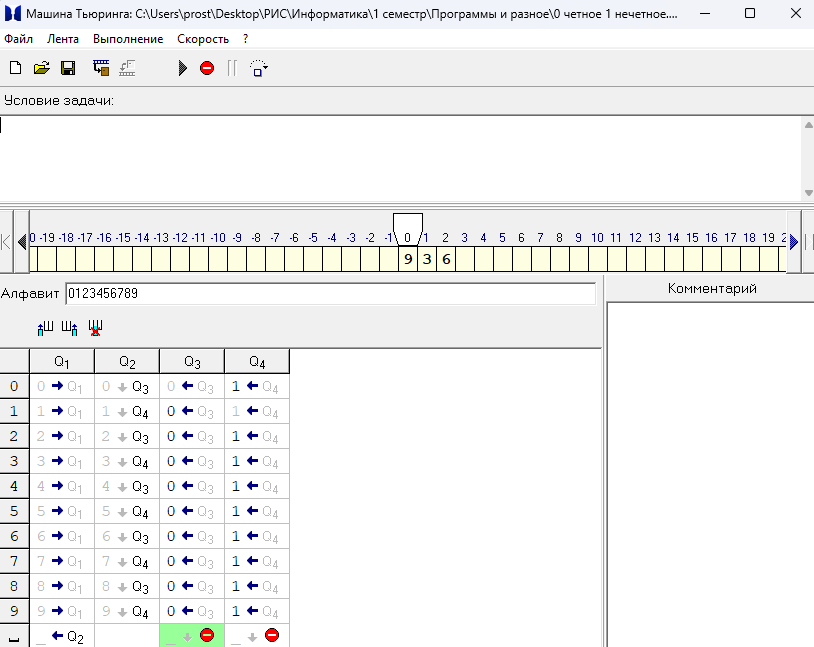
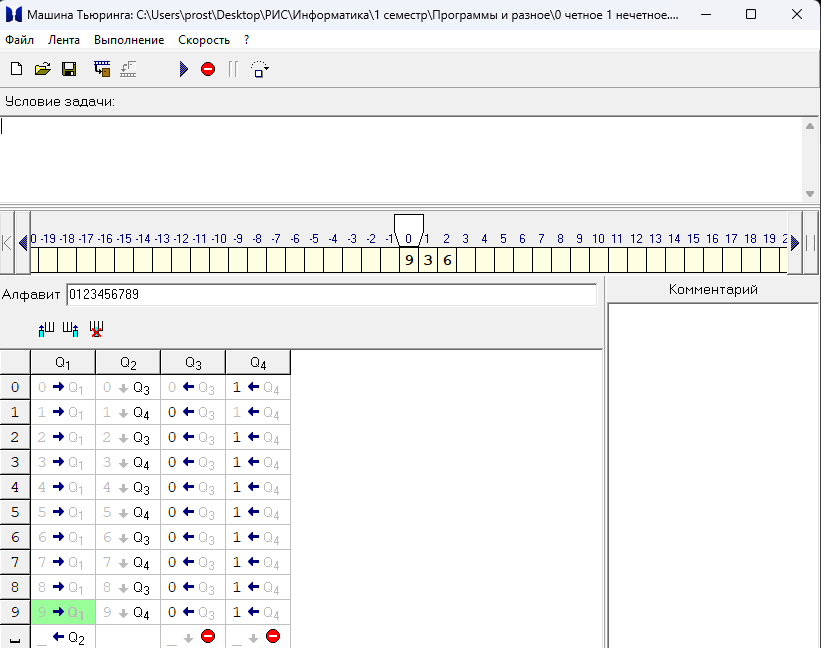
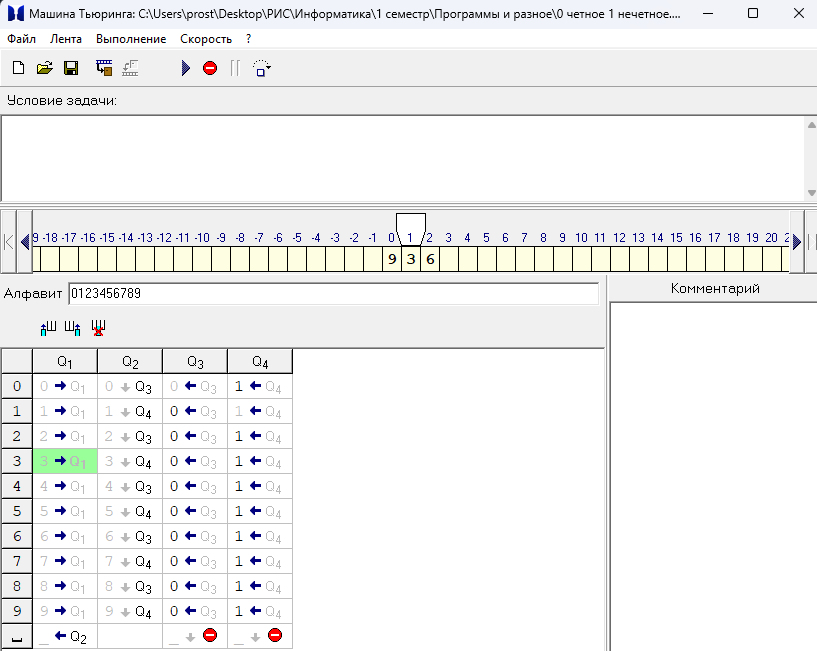
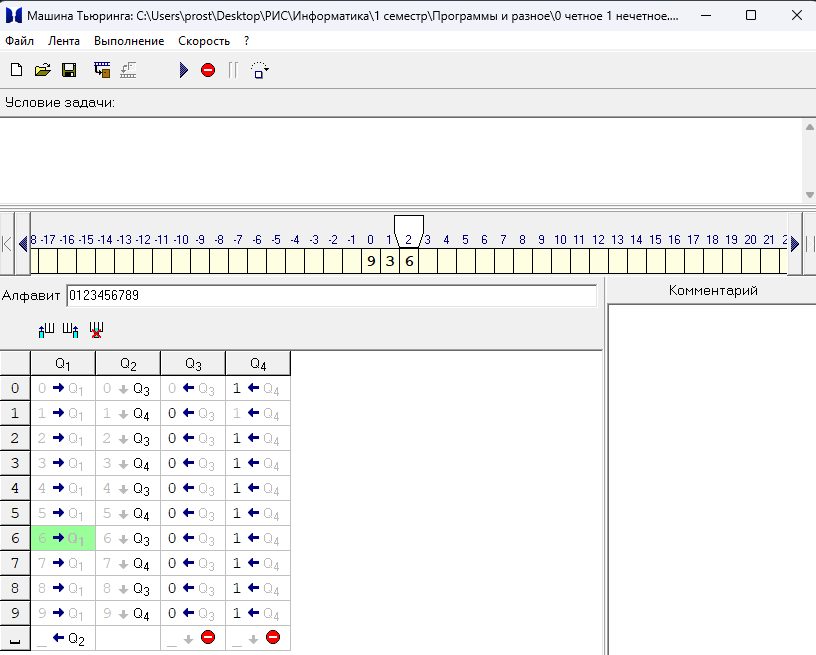
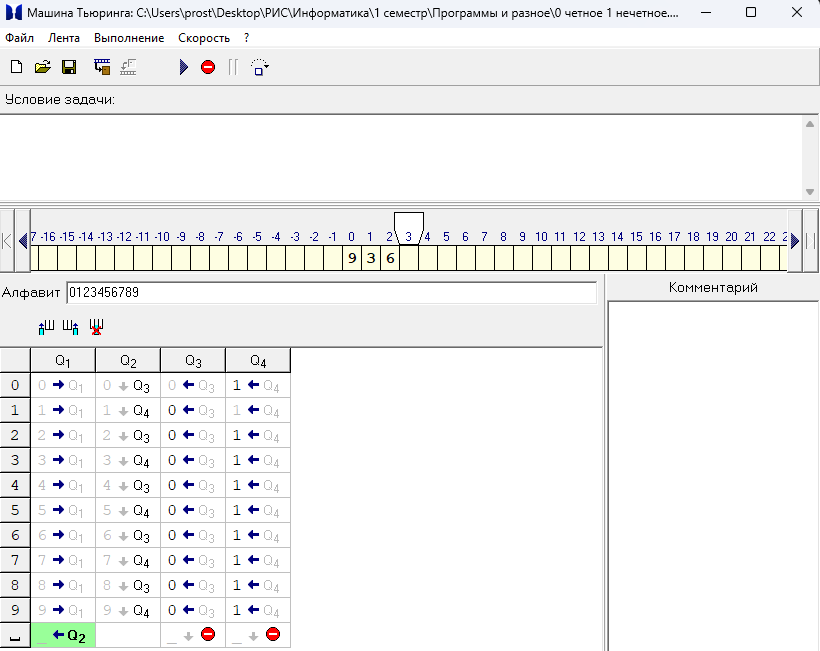


Рисунок 13 – тренажер МТ









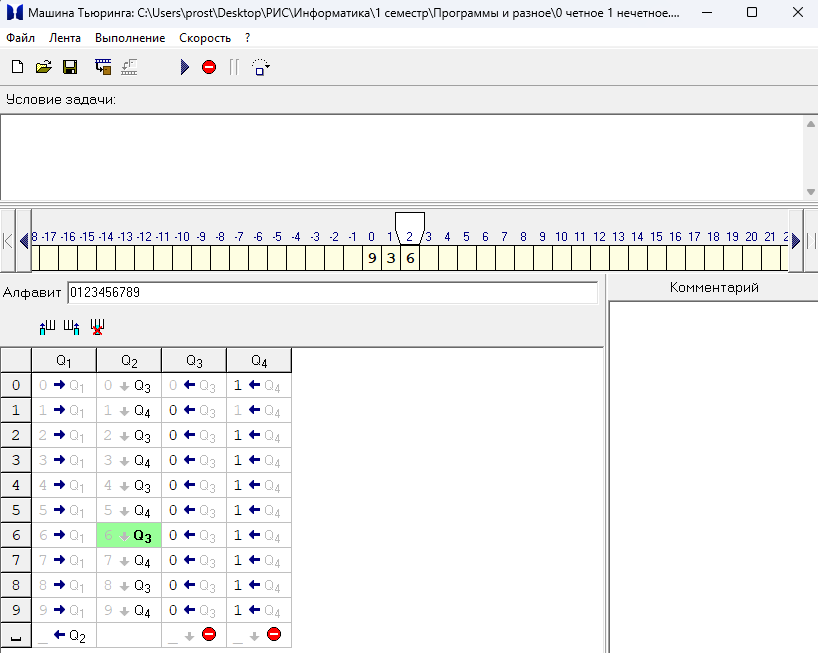


Рисунок 14 – смещение головки с помощью состояния Q1 и переход в состояние в Q2

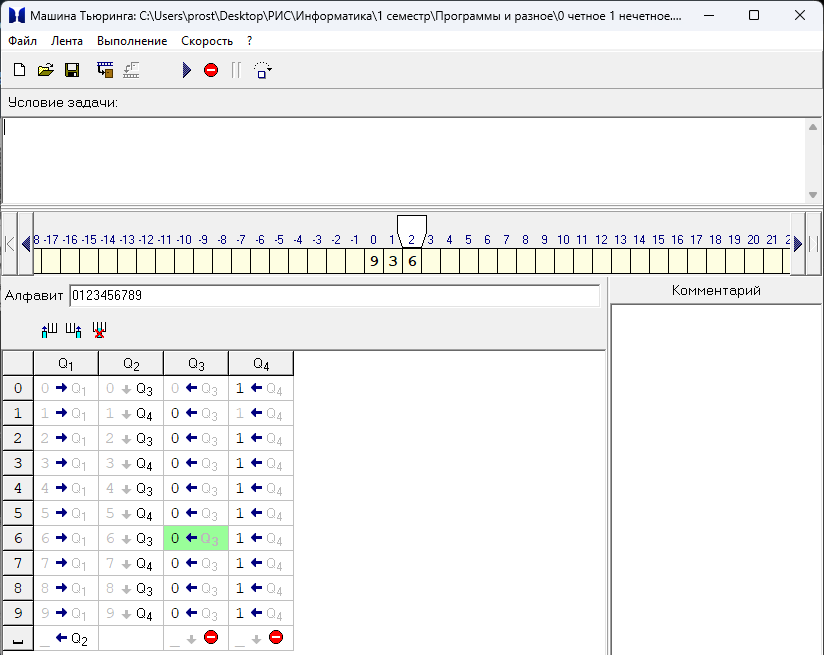
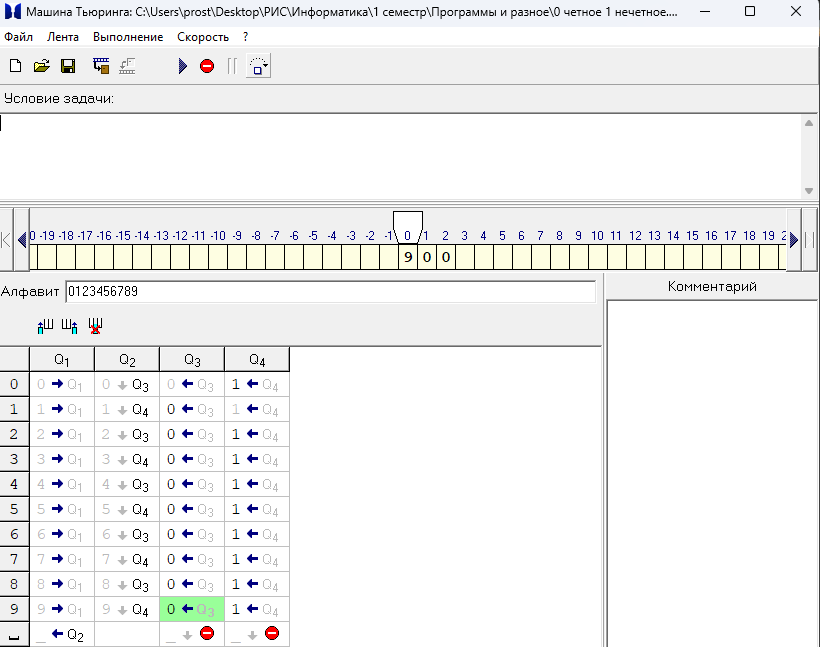
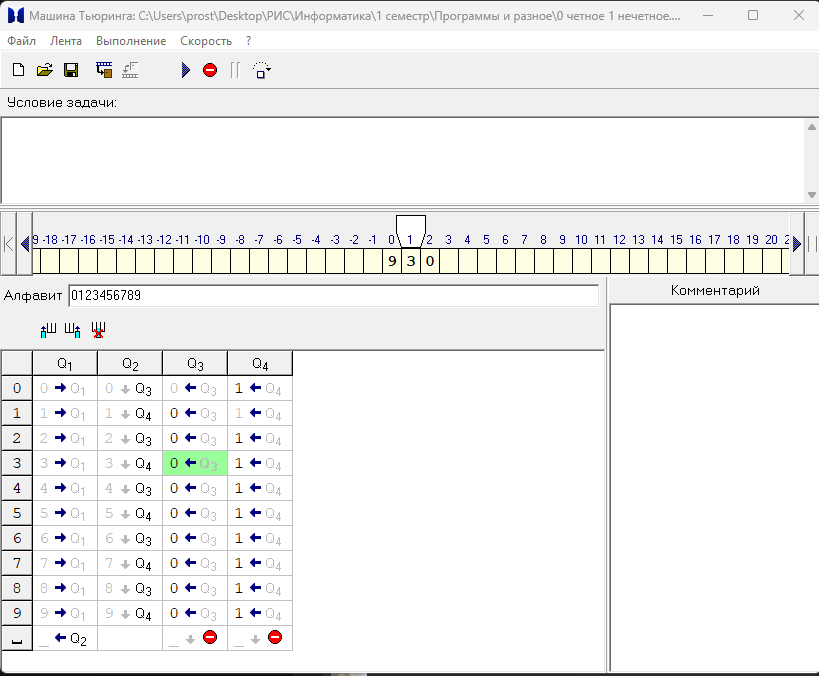
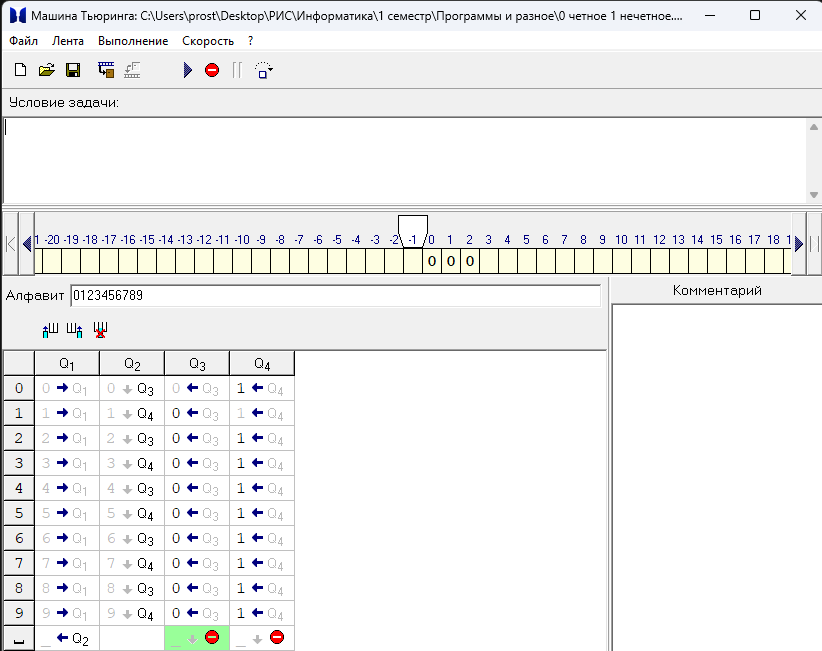


Рисунок 15 – переход из состояния Q2, в состояние Q3





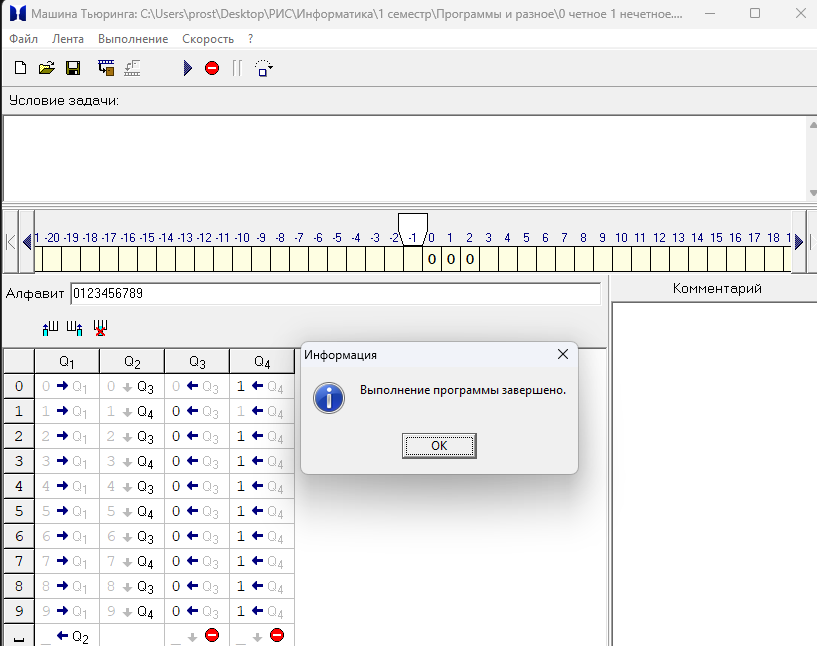


Рисунок 16 – замена чисел с помощью состояния Q3 и конец программы

**Алгоритм Маркова**

1. Постановка задачи №1

Дано: Непустое слово P Алфавит = {a, b}

Цель: удалить первый символ слова P. Пустое слово не менять

1. Правила:

1. \*a |→ .

2. \*b |→ .

3. \* |→ .

4. \_ → \*

Пусть дана строка: bba из которой должна получится строка: ba.

bba = (4) \*bba = (2) ba

1. Визуализация и проверка алгоритма в тренажере

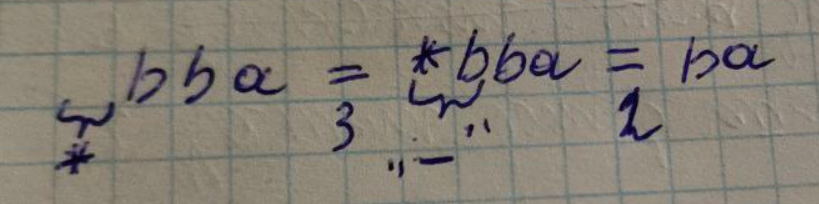


Рисунок 17 – визуализации алгоритма

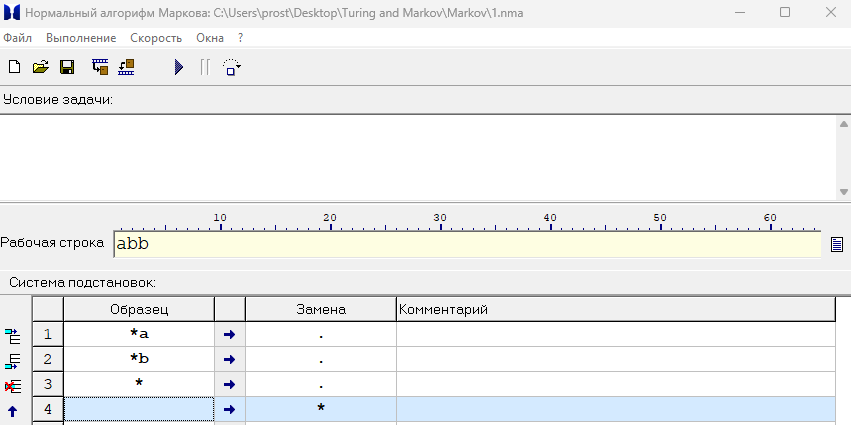
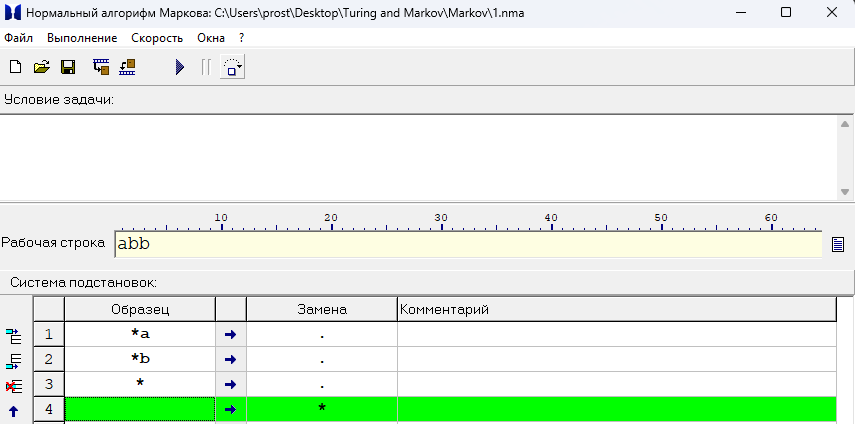
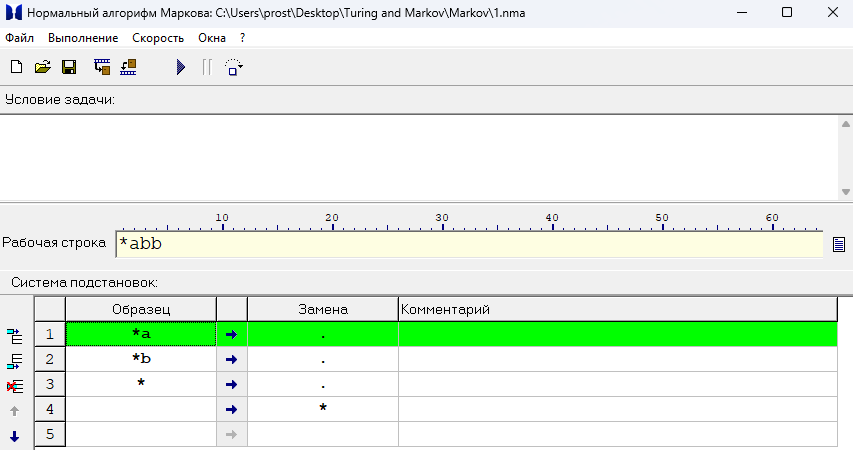


Рисунок 18 – тренажер алгоритма Маркова





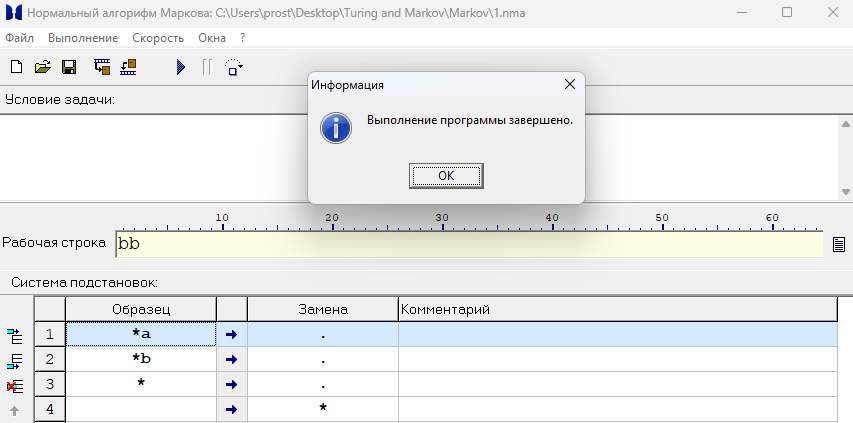


Рисунок 19 – демонстрация работы алгоритма

1. Постановка задачи №2

Дано: Непустое слово P Алфавит = {a, b, c, d}

Цель: в слове P требуется удалить все вхождения символа с, а затем заменить первое вхождение под слова bb на ddd.

1. Правила:

1. с → «\_»

2. bb |→ ddd.

Пусть дана строка: acabb из которой должна получится строка: aaddd.

acabb = (1) aabb = (2) aaddd

1. Визуализация и проверка алгоритма в тренажере

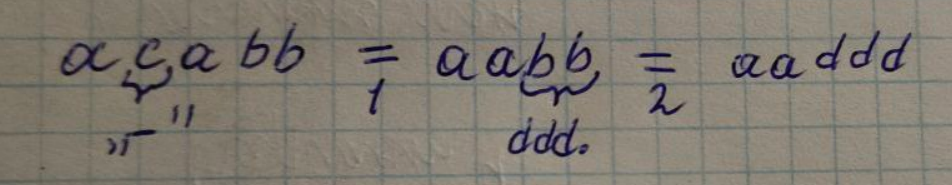


Рисунок 20 – визуализация алгоритма

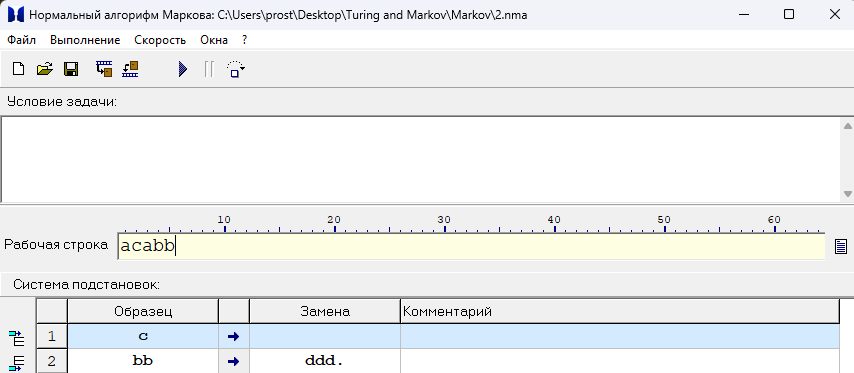
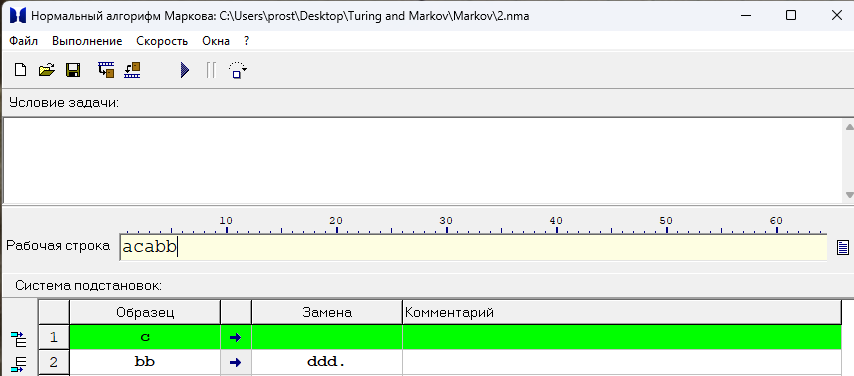
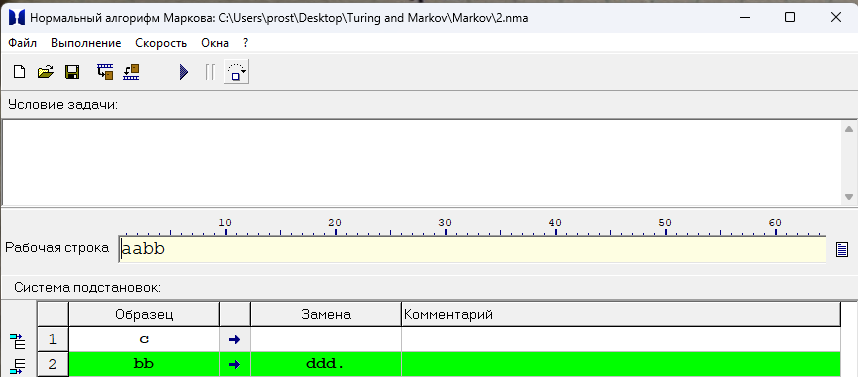


Рисунок 21 – тренажер алгоритмов Маркова





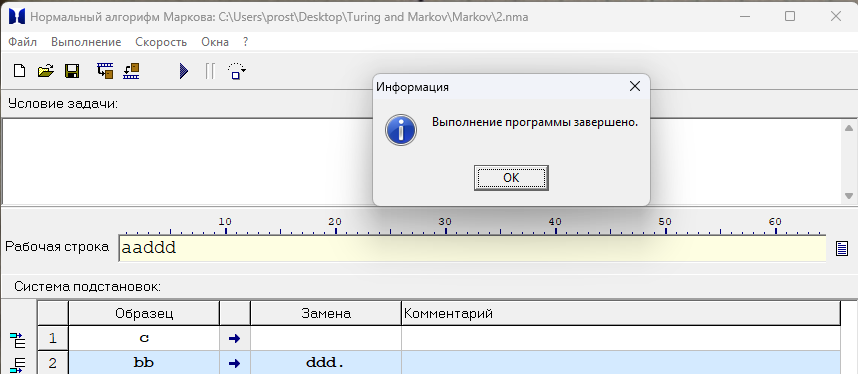


Рисунок 22 - демонстрация работы алгоритма

1. Постановка задачи №2

Дано: Непустое слово P Алфавит = {a, b,}

Цель: требуется приписать символ a к концу слова P

1. Правила:
2. \*a → a\*
3. \*b → b\*
4. b\* |→ ba.
5. a\* |→ aa.
6. \_ → \*

Пусть дана строка: abab из которой должна получится строка: ababa.

abab = (5) \*abab = (1) a\*bab = (2) ab\*ab = (1) aba\*b = (2) abab\* = (3) ababa

1. Визуализация и проверка алгоритма в тренажере

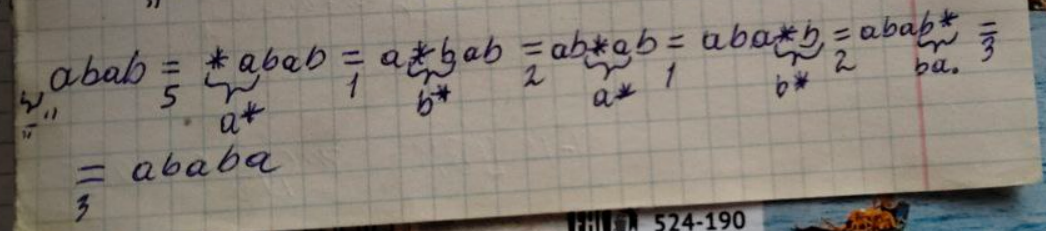


Рисунок 23 – визуализация алгоритма

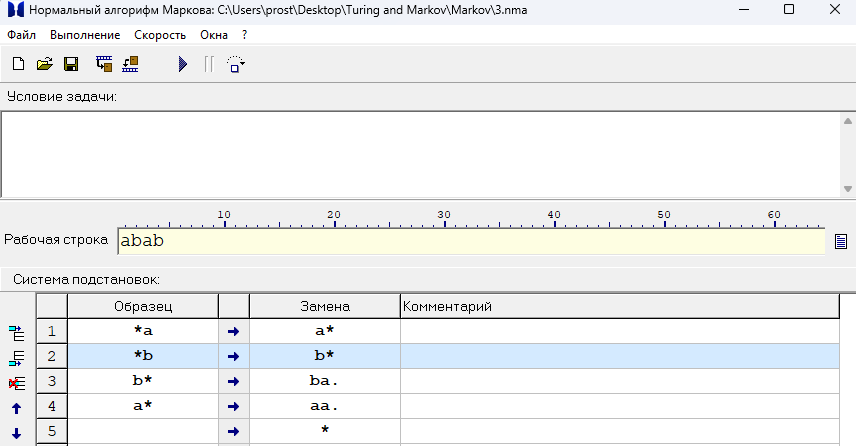
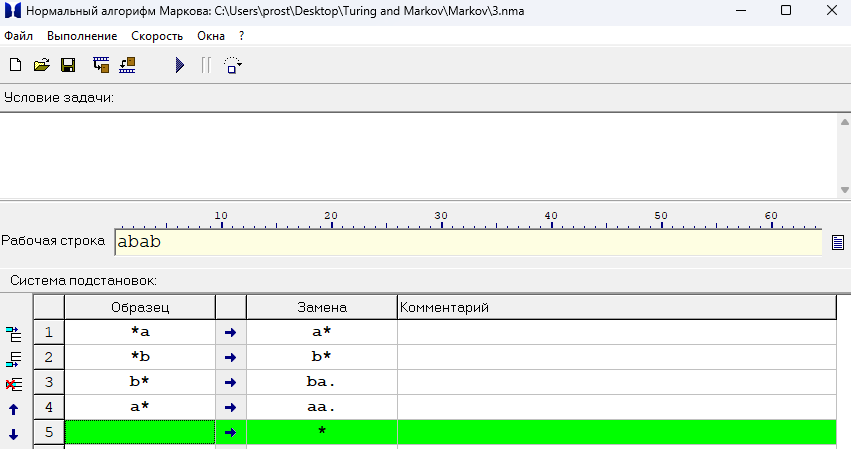
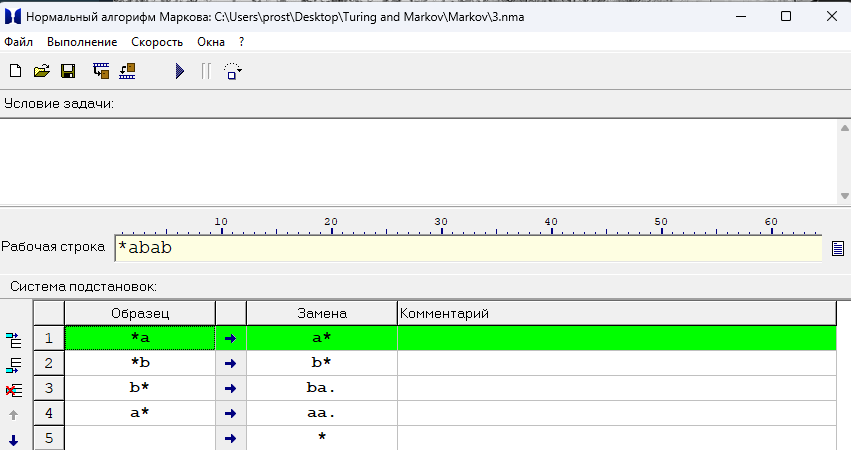
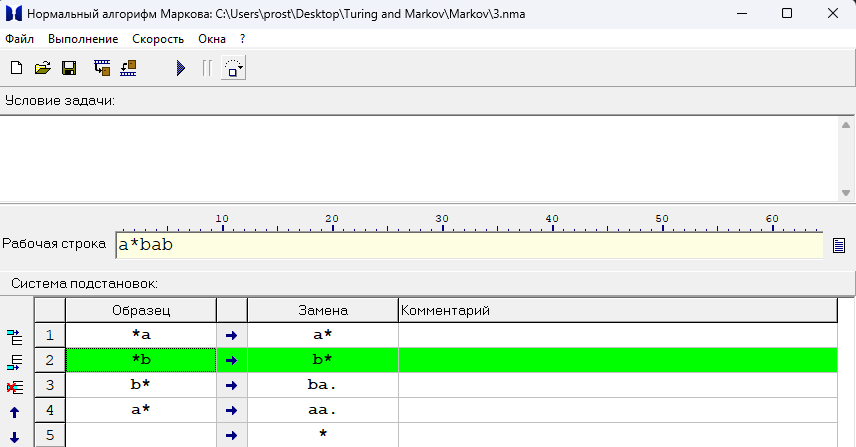
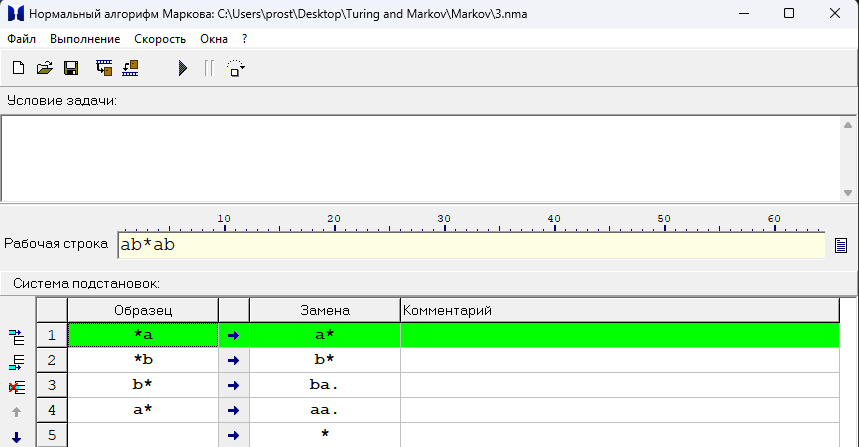


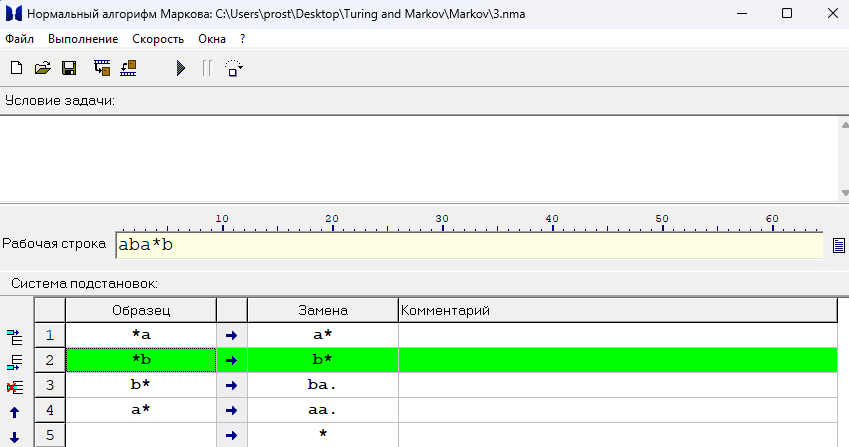
Рисунок 24 – тренажер алгоритма Маркова

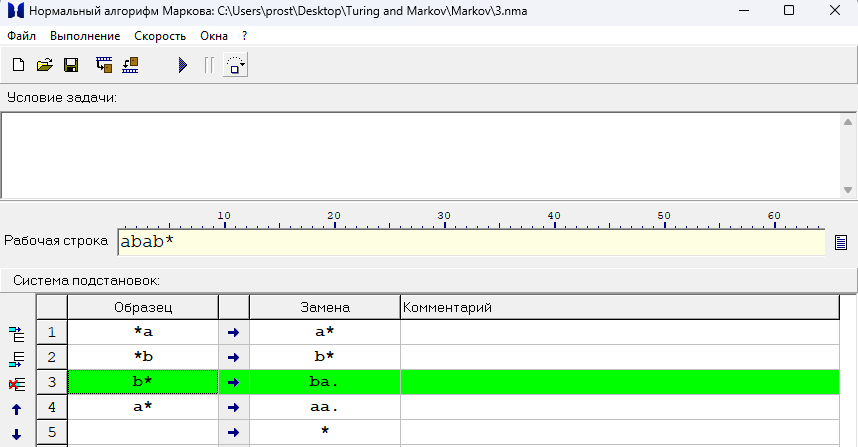












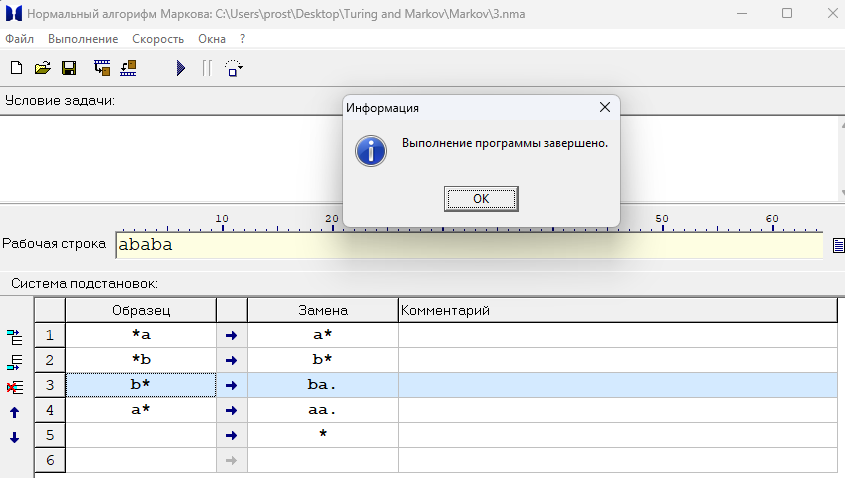


Рисунок 25 – демонстрация работы алгоритма