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»

**Лабораторная работа №1.**

**«Модели Машины Тьюринга и Алгоритмов Маркова»**

Выполнил студент гр. РИС-24-2б

Корпачев Матвей Егорович

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2024

**Машина Тьюринга**

* Машина Тьюринга состоит из 3-х частей:

1. Устройство управления.
2. Голова Машины.
3. Исполняющая лента.

* Элементарный шаг машины Тьюринга:

1. Голова считывает символ с ленты под ней.
2. Символ, который считан, и состояние головы обеспечивают переход машины Тьюринга в новое состояние  
   q1a1> q2 a2 d  
   Где:   
   a1 – символ, который считывается  
   q2 – новое состояние головы  
   a2 – новый записываемый символ  
   d - перемещение головы

**Решение задач**

**1.** Дано число, состоящее из 0 и 1. Заменить все 0 на 1 и 1 на 0.

Алгоритм решения:

1. Если под головой машины находится 0, тогда заменяем 0 на 1 и перемещаем голову влево, при этом состояние головы остаётся Q1   
2. Если под головой машины находится 1, тогда заменяем 1 на 0 и перемещаем голову влево, при этом состояние головы остаётся Q1   
3. Если под головой машины пустая ячейка, тогда завершаем выполнение программы.  
4. ↓ - это голова машины

|  |  |
| --- | --- |
|  | Q1 |
| 0 | 1<Q1 |
| 1 | 0<Q1 |
| - | . |

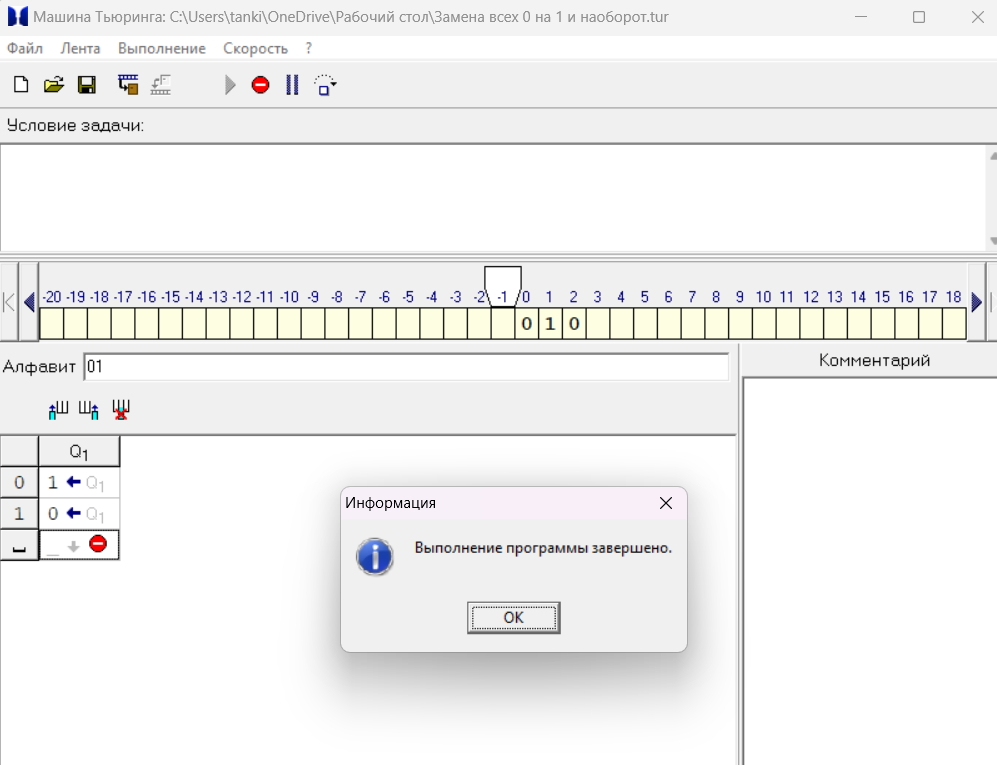
Команды:

Исходная строка:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ↓ |  |  |  |  |
|  |  |  | 1 | 0 | 1 |  |  |  |  |

Изменения:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | ↓ |  |  |  |  |  |
|  |  |  | 1 | 0 | 0 |  |  |  |  |
|  |  |  | ↓ |  |  |  |  |  |  |
|  |  |  | 1 | 1 | 0 |  |  |  |  |
|  |  | ↓ |  |  |  |  |  |  |  |
|  |  |  | 0 | 1 | 0 |  |  |  |  |

Скрин задания:  


**2.** На ввод подаётся случайное число. Прибавить 4.

Алгоритм решения:

1. Если под головой машины 0, 1, 2, 3, 4 или 5 заменяем на число, которое больше на 4 и завершаем выполнение программы. (0 на 4, 1 на 5, 2 на 6, 3 на 7, 4 на 8, 5 на 9)  
2. Таблица Q1. Если под головой машины 6, 7, 8 или 9, тогда надо заменить эти числа и перейти в следующий десяток. 6 заменяем на 0 и сдвигаем голову влево, при этом состояние головы сменяется на Q2. (аналогично заменяем 7 на 1, 8 на 2 и 9 на 3)  
3. Таблица Q2. Если под головой машины числа 0, 1, 2, 3, 4, 5, 6, 7 и 8, тогда заменяем их на число, которое больше исходного на один (повышает десяток) и завершает работу программы.   
Если под головой машины число 9, тогда заменяем 9 на 0 двигаемся влево, при этом состояние головы остаётся Q2  
4. Если под головой машины пустая ячейка и голова машины находится в состоянии Q2, тогда заполняем ячейку 1 и завершаем работу программы.

Команды:

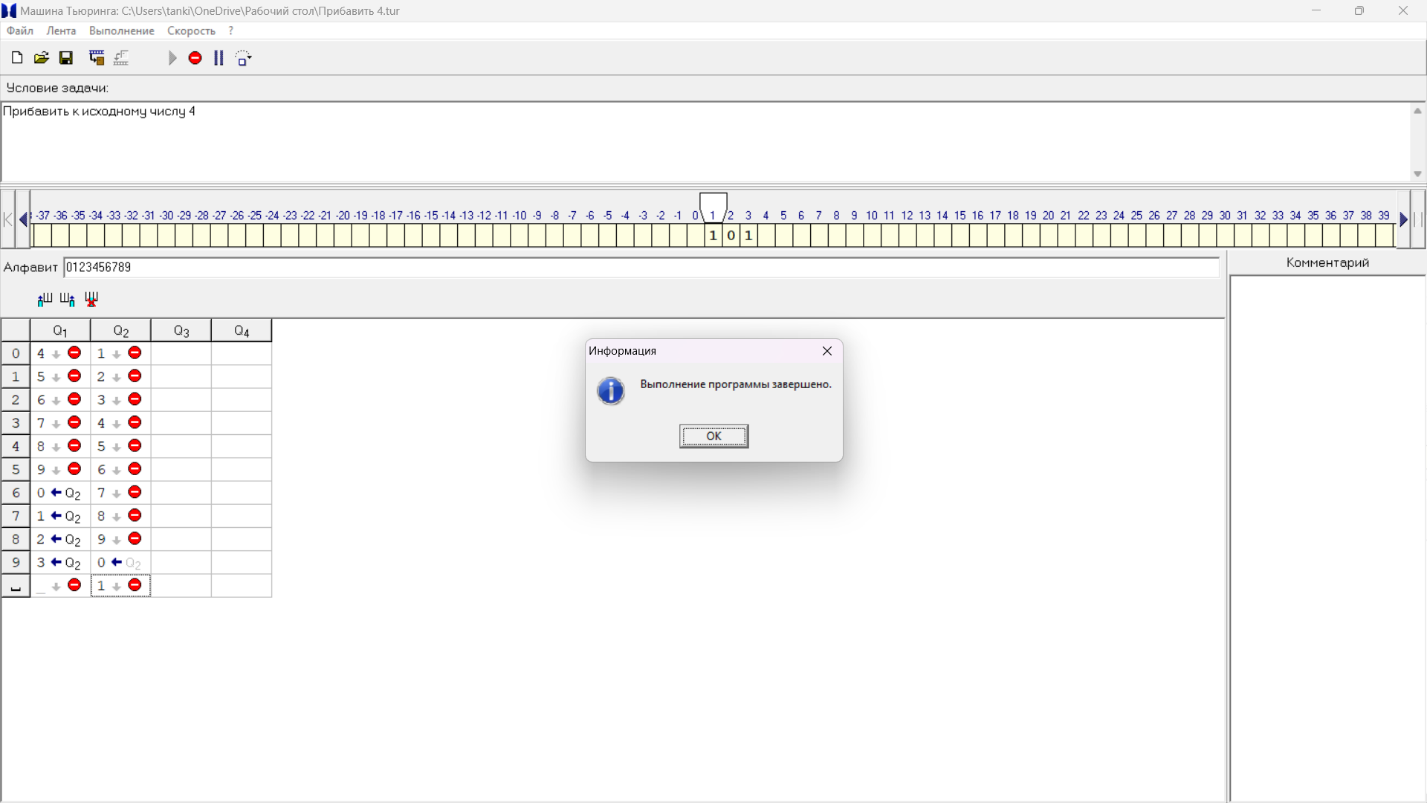
|  |  |  |
| --- | --- | --- |
|  | Q1 | Q2 |
| 0 | 4<. | 1<. |
| 1 | 5<. | 2<. |
| 2 | 6<. | 3<. |
| 3 | 7<. | 4<. |
| 4 | 8<. | 5<. |
| 5 | 9<. | 6<. |
| 6 | 0<Q2 | 7<. |
| 7 | 1<Q2 | 8<. |
| 8 | 2<Q2 | 9<. |
| 9 | 3<Q2 | 0<Q2 |
| - | . | 1. |

Исходная строка:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | ↓ |  |  |  |
|  |  |  |  |  | 9 | 7 |  |  |  |

Изменения:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ↓ |  |  |  |  |
|  |  |  |  |  | 9 | 1 |  |  |  |
|  |  |  |  | ↓ |  |  |  |  |  |
|  |  |  |  |  | 0 | 1 |  |  |  |
|  |  |  |  | ↓ |  |  |  |  |  |
|  |  |  |  | 1 | 0 | 1 |  |  |  |

Скрин задания:  
  
  
**3.** На ввод подаётся случайное число. Если число чётное – заменить в нём все цифры на 0, иначе на 1.

Если число чётное, значит оно всегда будет оканчиваться чётной цифрой. Если число нечётное, значит оно всегда будет оканчиваться нечётной цифрой.

Алгоритм решения:

1. Если под голой машины чётное число, тогда заменить число на 0 и сдвинуть голову влево, сменить состояние головы на Q2. Далее любое число под головой машины заменяется на 0.   
2. Если под голой машины нечётное число, тогда заменить число на 1 и сдвинуть голову влево, сменить состояние головы на Q3. Далее любое число под головой машины заменяется на 1.  
3. Если под головой машины пустая ячейка, тогда завершаем выполнение программы.

Команды:

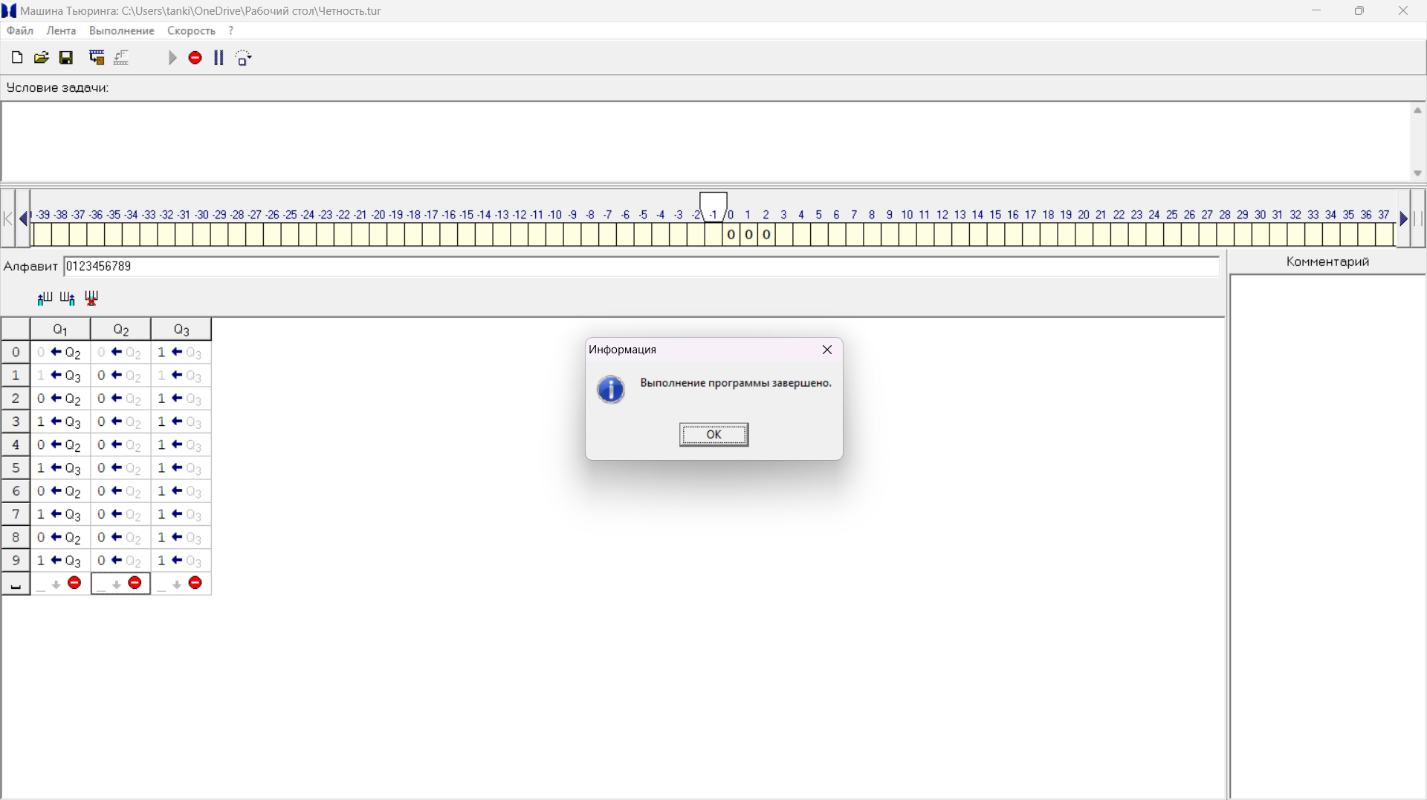
|  |  |  |  |
| --- | --- | --- | --- |
|  | Q1 | Q2 | Q3 |
| 0 | 0<Q2 | 0<Q2 | 1<Q3 |
| 1 | 1<Q3 | 0<Q2 | 1<Q3 |
| 2 | 0<Q2 | 0<Q2 | 1<Q3 |
| 3 | 1<Q3 | 0<Q2 | 1<Q3 |
| 4 | 0<Q2 | 0<Q2 | 1<Q3 |
| 5 | 1<Q3 | 0<Q2 | 1<Q3 |
| 6 | 0<Q2 | 0<Q2 | 1<Q3 |
| 7 | 1<Q3 | 0<Q2 | 1<Q3 |
| 8 | 0<Q2 | 0<Q2 | 1<Q3 |
| 9 | 1<Q3 | 0<Q2 | 1<Q3 |
| - | . | . | . |

Исходная строка:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ↓ |  |  |  |  |
|  |  |  | 9 | 7 | 2 |  |  |  |  |

Изменения:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | ↓ |  |  |  |  |  |
|  |  |  | 9 | 7 | 0 |  |  |  |  |
|  |  |  | ↓ |  |  |  |  |  |  |
|  |  |  | 9 | 0 | 0 |  |  |  |  |
|  |  | ↓ |  |  |  |  |  |  |  |
|  |  |  | 0 | 0 | 0 |  |  |  |  |

Скрин задания:  


**Нормальные алгорифмы Маркова**

* Модель алгорифмов Маркова преобразует исходные слова с помощью заданных правил, которые могут заменять часть исходной строки.
* Слова – это строки, состоящие из символов.
* Также в алгорифмах Маркова используются приоритеты применения этих правил:  
  1. Правило находит первое своё вхождение.  
  2. Правила выполняются по порядку их номеров.  
  3. Если правило не применимо к данной строке, то алгоритм переходит к выполнению следующего по порядку правила.  
  4. Если правило выполнено, то алгоритм возвращается к проверке правил, снова начиная с начала списка.  
  5. Алгоритм завершается, если не одно правило не применимо.
* Виды подстановок:  
  1. Терминальная – это подстановка, после которой выполнение алгоритма заканчивается.  
  2. Нетерминальные – это подстановка, которая предполагает дальнейшее применение правил.

**Решение задач**

**1.** A={a, b}. Удалить из непустого слова P его первый символ. Пустое слово не меня.

Правила:

1. \*b > . (Терминальная замена)
2. \*a > . (Терминальная замена)
3. > \* (Нетерминальная замена)

Пусть исходная строка: abbbaab

1-й шаг. Сначала программа выполнит 3-е правило - добавит ‘\*’ в начало строки. (Именно 3, так как подстрок ‘\*a’ и ‘\*b’ в строке ещё нет)

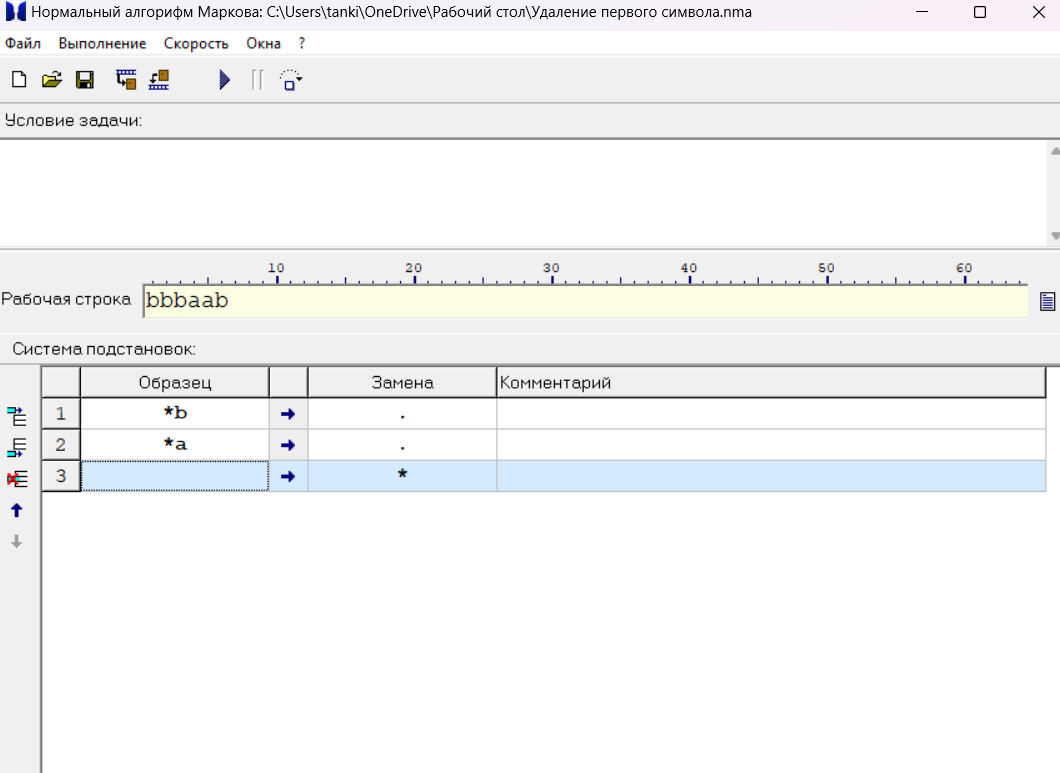
Изменения: abbbaab > \*abbbaab

2-й шаг. Далее выполнит 1-е или 2-е правило, всё зависит от первого символа в исходной строке. Если первый символ в исходной строке ‘a’, то выполнится 2-е правило - программа удалит ‘\*a’ и завершит свою работу. Если первый символ в исходной строке ‘b’, то выполнится 1-е правило - программа удалит ‘\*b’ и завершит свою работу.

Изменения: \*abbbaab > bbbaab

Итог: Первый символ непустого слова P удалён.

Скрин задания:



**2.** A={a, b, c, d}. В слове P требуется удалить все вхождения символа ‘c’, а затем заменить первое вхождение подслова ‘bb’на ‘ddd’.

Правила:

1. с >
2. bb > ddd.

Пусть исходная строка: abbcdbcbca

1-й шаг. Программа выполнит 1-е правило - удалит все символы ‘с’ в слове P.

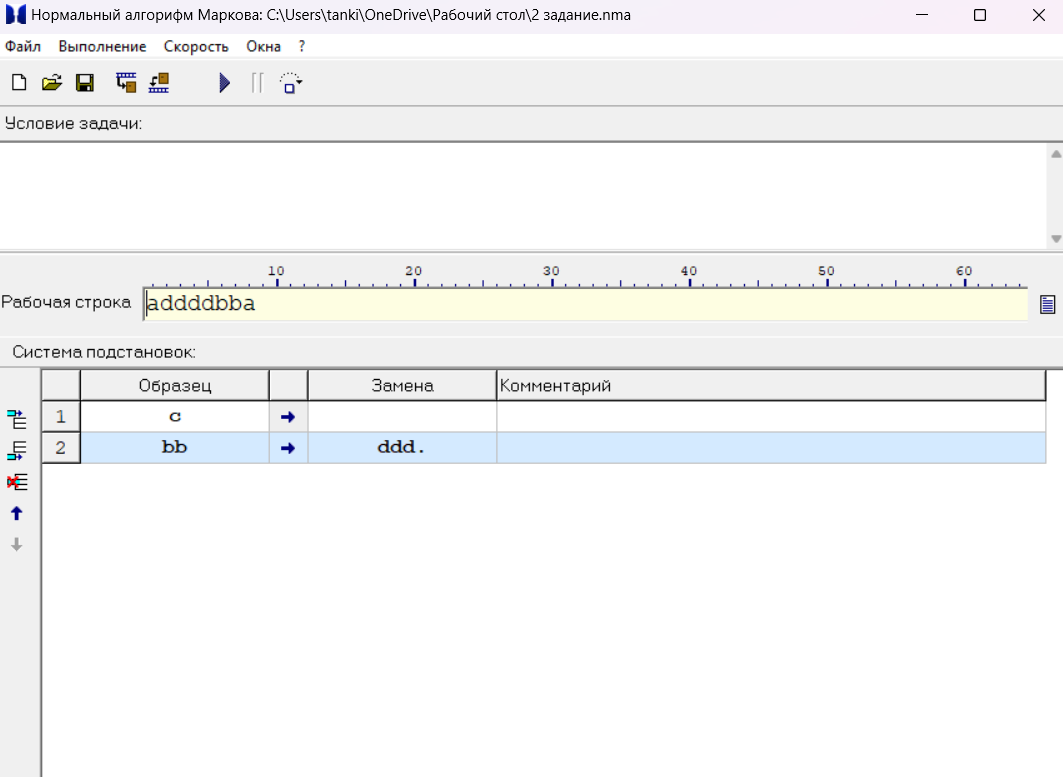
Изменения: abbcdbcbca > abbdbсbсa > abbdbbca > abbdbba

2-й шаг. Программа выполнит 2-е правило - заменит первое вхождение подслова ‘bb’ на ‘ddd’ и завершит свою работу.

Изменения: abbdbba > addddbba

Итог: В слове P удалены все вхождения символа ‘c’, первое вхождение подслова ‘bb’ заменено на ‘ddd’.

Скрин задания:



**3.** А={a, b}. Требуется приписать символ ‘a’ к концу слова P.

Правила:

1. \*a > a\*
2. \*b > b\*
3. \* > a.
4. > \*

Пусть исходная строка: abba

1-й шаг. Программа выполнит 4 правило – добавит ‘\*’ в начало строки. (4 правило, так как подстрок ‘\*a’ и ‘\*b’ и символа ‘\*’ в строке ещё нет)

Изменения: abba > \*abba

2-й шаг. Программа выполнит 1-е и 2-е правила. Таким образом звёздачка переместиться в конец строки.

Изменения: \*abba > a\*bba > ab\*ba > abb\*a > abba\*

3-й шаг. Программа больше не может выполнить 1-е и 2-е правило, подстрок ‘\*a’ и ‘\*b’ в строке больше нет. Следовательно, выполнится 3-е правило. Символ ‘\*’ заменится на ‘а’ и программа завершит свою работу.

Изменения: abba\* > abbaa

Итог: Символ ‘а’ приписан к концу слова P.

Скрин задания:

