Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

**Факультет:** электротехнический

**Направление:** 09.03.04 Программная инженерия

ОТЧЁТ

По теме «Алгоритм Маркова»

Дисциплина: основы алгоритмизации и программирования

Выполнил:

студент группы РИС-21-1б

Кудрявцева Рената Максимовна

Проверила:

доцент кафедры ИТАС

О. А. Полякова

Пермь, 2021

**Содержание**

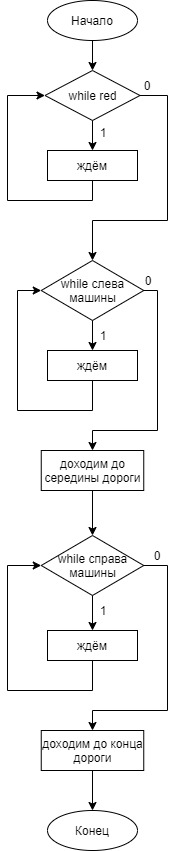
|  |  |
| --- | --- |
| 1. Словесный алгоритм перехода перекрёстка со светофором……….......... | 3 |
| 1. Алгоритм Маркова |  |
| А) Задача №1………………………………………………………………… | 4 |
| Б) Задача №2……………………………………………………………….... | 6 |
| В) Задача №3……………………………………………………………........ | 8 |
| Г) Задача №4……………………………………………………………........ | 10 |
|  |  |
|  |  |

**Словесный алгоритм перехода перекрёстка со светофором**

**Постановка задачи**

Составить блок-схему к словесному алгоритму перехода перекрёстка со светофором.

**Блок-схема**



**Алгоритм Маркова**

Для решения всех приведённых ниже задач можно использовать метод **replace()**. Данный метод возвращает копию строки, в которой все вхождения подстроки заменяются.

Синтаксис: s.replace(old, new, count)

Old – старая подстрока, которую необходимо заменить

New – новая опдстрока, которая заменяет старую

Count – сколько раз нужно заменить подстроку (необязательный параметр)

**Задача №1**

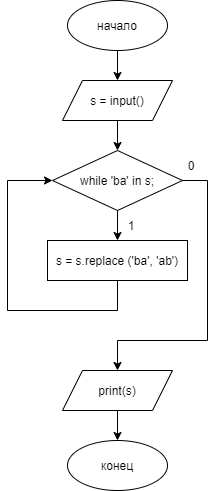
**Постановка задачи**

Дан алфавит: a, b. Составить правила преобразования, по которым исходное слово приводится к виду “aaaaabbbb” – все “а” слева, а “b” справа.

**Анализ**

Задача решается при помощи одной формулы: ba → ab. Пока в слове справа хотя бы от одного символа b есть символ a , эта формула будет переносить a налево от этого b. Формула перестанет работать, когда справа от b нет ни одного a.

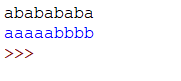
**Блок-схема**



**Код Python**

s = input()  
while 'ba' in s:  
 s = s.replace('ba', 'ab')  
print(s)

**Скриншот результата**



**Задача №2**

**Постановка задачи**

А={a,b}. Удалить из непустого произвольного слова Р его первый символ. Пустое слово не менять.

**Анализ задачи**

Формулы для решения задачи:  
1. \*а|→

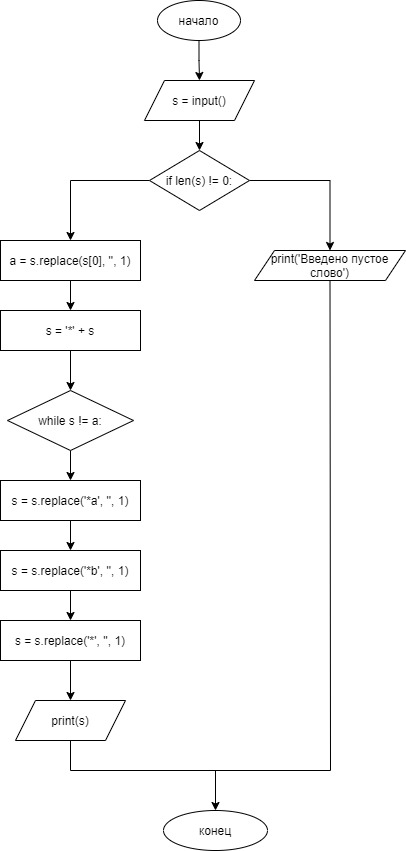
2. \*b|→

3. \* |→

4. → \*

Звёздочка нужна для того, чтобы обозначить первый символ. После добавления звёздочки слева \* с 1 символом будут удалены. Чтобы алгоритм не зациклился, слева нужно добавить \*.

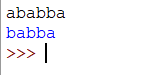
**Блок-схема**



**Код Python**

s = input()  
if len(s) != 0:  
 a = s.replace(s[0], '', 1)  
 s = '\*' + s  
 while s != a:  
 s = s.replace('\*a', '', 1)  
 s = s.replace('\*b', '', 1)  
 s = s.replace('\*', '', 1)  
 print(s)  
else:  
 print('Введено пустое слово')

**Скриншот результата**



**Задача №3**

**Постановка задачи**

А={a,b,c,d}. В произвольном слове Р требуется удалить все вхождения символа c, затем заменить первое вхождение подслова bb на ddd.

Например, abbcabbca —> adddabba

**Анализ задачи**

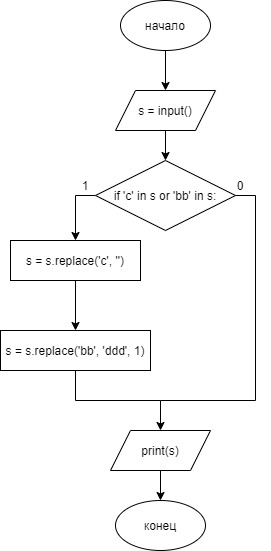
Формулы:

1. c →

2. bb |→ ddd

Сначала алгоритм удаляет все вхождения c, после этого заменяет bb один раз и заканчивает выполнение.

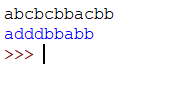
**Блок-схема**



**Код Python**

s = input()  
if 'bb' in s or 'c' in s:  
 s = s.replace('c', '')  
 s = s.replace('bb', 'ddd', 1)  
print(s)

**Скриншот результата**



**Задача №4**

**Постановка задачи**

А={a,b}. Требуется приписать символ «a» к концу произвольного слова Р.

**Анализ задачи**

Формулы:

1. \*a → a\*

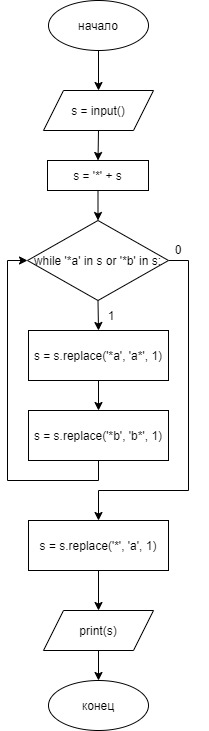
2. \*b → b\*

3. \* |→ a

4. →\*

Ставим \* слева и перемещаем её в конец. Далее заменяем эту \* на символ a и заканчиваем выполнение алгоритма.

**Блок-схема**



**Код Python**

s = input()  
s = '\*' + s  
while '\*a' in s or '\*b' in s:  
 s = s.replace('\*a', 'a\*', 1)  
 s = s.replace('\*b', 'b\*', 1)  
s = s.replace('\*', 'a', 1)  
print(s)

**Скриншот результата**

