

Дискретная математика. Задание №2.

**Условие задачи:** Составьте схему нормального алгоритма Маркова для решения одной из предложенных задач по выбору. Приведите протокол решения тестового примера из описания выбранного варианта.

**Выбранный вариант №3:**

Составьте схему нормального алгоритма Маркова для определения, каких символов "a" или "b" больше в слове в алфавите {a,b} в виде слова из избыточных символов большего количества.

Например, для слова "abaabaab" результатом будет слово "aa", так как в слове 5 символов "a" и 3 символа "b" и избыток "a" над "b" будет 2. В случае одинакового числа символов "a" и "b" результатом будет пустое слово.

Например, результатом для слова "babbaa" будет пустым.

Приведите протокол вычислений для слова "abaabaab".

**Решение задачи:**

Алфавит содержит символы "a", "b".

Правила представлены следующим образом:

ab ->

ba ->

Исходное состояние = "abaabaab".

Шаг 0. "abaabaab"

Шаг 1. "aabaab"

Шаг 2. "aaab"

Шаг 3. "aa"

**Проверка произведённых вычислений:**

В исходном слове "abaabaab" символ "a" встречается 5 раз, а символ "b" - 3 раза.

Следовательно, символов "a" в исходном слове на 2 штуки больше.

На последнем шаге работы нормального алгоритма Маркова получено "aa", что является верным ответом.

Ко всему прочему, был реализован программный код на языке C++ для проверки реализованных правил:

```
#include <cmath>
#include <iostream>
#include <string>

void Markov(std::string& source) {
    const std::string ab{ "ab" };
    const std::string ba{ "ba" };
    size_t pos_ab{ source.find(ab) };
    size_t pos_ba{ source.find(ba) };

    while(pos_ab != std::string::npos || pos_ba != std::string::npos) {
        source.erase(std::min(pos_ab, pos_ba), ab.length());
        pos_ab = source.find(ab);
        pos_ba = source.find(ba);
    }
}

int main() {
    // Прочитать исходное слово от пользователя из стандартного ввода.
    std::string word;
    std::cout << "Алфавит = a, b\n"
                "Правила = ab ->\n"
                "          ba ->\n\n"
                "Введите исходное слово: ";
    std::cin >> word;
    std::cout << "\nИсходное слово: " << word << '\n';

    // Проверить что слово состоит только из символов алфавита, то есть из "a" и "b".
    if(word.find_first_not_of("ab") != std::string::npos) {
        std::cerr << "\n\nВведено некорректное слово.\n"
                    << "Исходный алфавит содержит только символы \"a\" и \"b\".\n";
        return 1;
    }

    // Прежде чем воспользоваться вычислениями по алгоритму Маркова, проверим, что у нас должно выйти.
    // 1. Подсчитаем количество символов "a" и "b" в исходном слове.
    auto count_a{ std::count(word.begin(), word.end(), 'a') };
    auto count_b{ std::count(word.begin(), word.end(), 'b') };
    // 2. Результатом должно быть слово из избыточных символов большего количества.
    const std::string expected(std::abs(count_a - count_b), ((count_a > count_b) ? 'a' : 'b'));

    // Выполним алгоритм Маркова, вызвав описанную выше функцию.
    Markov(word);
    std::cout << "После применения алгоритма Маркова: " << word << '\n';

    // Проверим то, что полученное слово совпадает с ожидаемым.
    if(word != expected) {
```

```

std::cerr << "\n\nАлгоритм Маркова отработал некорректно."
        << "\nПолучено: " << word
        << "\nОжидается: " << expected << '\n';
return 1;
}
}

```

#### Проведение серии тестов:

```

> clang++ -std=c++2b markov.cpp
> ./a.out
Алфавит = a, b
Правила = ab ->
          ba ->

Введите исходное слово: abaabaab

Исходное слово: abaabaab
После применения алгоритма Маркова: aa
>
> ./a.out
Алфавит = a, b
Правила = ab ->
          ba ->

Введите исходное слово: a

Исходное слово: a
После применения алгоритма Маркова: a
>
> ./a.out
Алфавит = a, b
Правила = ab ->
          ba ->

Введите исходное слово: ab

Исходное слово: ab
После применения алгоритма Маркова:
>
> ./a.out
Алфавит = a, b
Правила = ab ->
          ba ->

Введите исходное слово: abbbbbbabbbbaaaabb

Исходное слово: abbbbbbabbbbaaaabb
После применения алгоритма Маркова: bbbbbb
~/code/spbpu/discrete 19s > █

```

Код успешно выполняет задачу в различных тестовых сценариях. Отсутствие вывода сообщения "Алгоритм Маркова отработал некорректно" свидетельствует о том, что условие `if(word != expected)` никогда не выполняется, что подтверждает корректность работы программы.