

Задача 1. Дан текст, содержащий от 2 до 30 слов, в каждом из которых от 2 до 15 латинских букв, между соседними словами не менее одного разделителя (. , : ; пробел, -).

Распечатать слова текста, отличные от последнего, предварительно преобразовав слово по следующему правилу:

- переставить первую букву в конец слова;
- удалить повторяющиеся буквы.

```
#include <iostream>
#include <fstream>
#include <string>
```

```
const int n = 30; // количество слов
const int m = 15; // длина слова
```

```
const std::string token(" ,!.:;-\\n");
```

```
void create_matrix(std::ifstream& file, std::string matrix[], int &row)
{
    auto skip_token = [&](char &c)
    {
        while (!file.eof() && token.find(c) != std::string::npos)
            c = file.get();
    };
    auto read_word = [&](char& c)->std::string
    {
        std::string result = "";
        result.reserve(m);
        while (!file.eof() && token.find(c) == std::string::npos)
        {
            result += c;
            c = file.get();
        }
        return result;
    };
    row = 0;
    char c = file.get();
    while (!file.eof() && row < n)
    {
        skip_token(c);
        if (token.find(c) == std::string::npos)
        {
            matrix[row] = read_word(c);
            if (matrix[row] != "")
                ++row;
        }
    }
}
```

```
void print_matrix(std::string matrix[], int row, const char* message)
{
    std::cout << message << '\\n';
    for (int i = 0; i < row; ++i)
        std::cout << '-' << matrix[i] << '-' << '\\n';
}
```

```
// переставить первую букву в конец слова
```

```
void change1(std::string &word)
{
    word += word[0];
    word.erase(word.begin());
}
```

```

void change2(std::string& word)
{
    for (size_t i = 1; i < word.length();
        if (word.substr(0, i).find(word[i]) != std::string::npos)
            word.erase(word.begin() + i);
        else
            ++i;
    }

void task1(std::string matrix[], int row)
{
    for (int i = 0; i < row - 1; ++i)
        if (matrix[i] != matrix[row - 1])
        {
            change2(matrix[i]);
            std::cout << matrix[i] << '\n';
        }
}

int main()
{
    std::ifstream file("text.txt");
    if (!file)
    {
        std::cout << "File error\n";
    }
    else
    {
        std::string matrix[n];
        int row = 0;
        create_matrix(file, matrix, row);
        print_matrix(matrix, row, "Original matrix:\n");
        task1(matrix, row);

        file.close();
    }

    std::cin.get();
    return 0;
}

```