Задача 2. 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, -).
Найти слова, оканчивающиеся на заданное окончание. Упорядочить найденные слова по возрастанию.

```
#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_matrix2(std::ifstream& file, std::string matrix[], int &row, std::string end)
{
	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].length() >= end.length() &&
			    matrix[row].find_last_of(end) == matrix[row].length() - 1)
				++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';
}

void sorting(std::string matrix[], int row)
{
	for (int count = row; count > 1; --count)
		for (int i = 0; i < count - 1; ++i)
			if (matrix[i] > matrix[i + 1])
				matrix[i].swap(matrix[i + 1]);
}
```

```
int main()
{
	std::ifstream file("text_ing.txt");
	if (!file)
	{
		std::cout << "File error\n";
	}
	else
	{
		std::string matrix[n];
		int row = 0;
		std::string end;
		std::cout << "Input the ending:\n";
		std::cin >> end; std::cin.ignore();
		create_matrix2(file, matrix, row, end);
		sorting(matrix, row);
		print_matrix(matrix, row, "Sorting matrix:");

		file.close();
	}

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