Задача 1. Списки

Задание: создать список по типу очереди из вводимых слов. Удалить слова содержащие букву «а» и посчитать количество удаленных слов. При завершении программы освободить динамическую память.

Выполнение: структура списка (в динамической памяти) изображена на рисунке 1

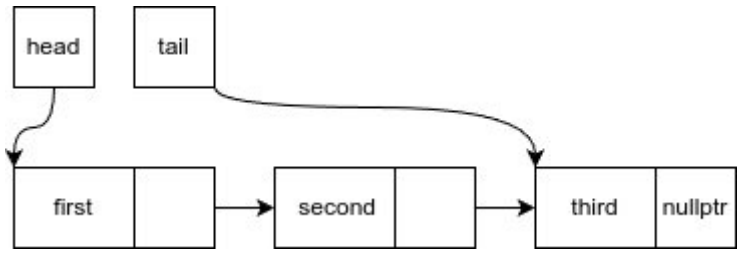

Рисунок 1 — Структура списка

Код программы:

```
#include <iostream>
using namespace std;

struct q_elem {
    string word;
    q_elem* next = nullptr;
};

int main() {
    cout << "Enter amount of elements: ";
    int N;
    cin >> N;

    if (N <= 0) {
        cout << "Amount of elements must be > 0!\n";
        return -1;
    }

    q_elem* head = nullptr;
    q_elem* tail = nullptr;
```

```
for (int i = 0; i < N; ++i) {
    auto* new_elem = new q_elem();
    cin >> new_elem->word;

    if (head == nullptr) {
        head = new_elem;
    } else {
        tail->next = new_elem;
    }
    tail = new_elem;
}

int counter = 0;
q_elem* p = head;
q_elem* prev = nullptr;

while (p != nullptr) {
    if (p->word.find('a') != string::npos) {
        if (p == head) {
            head = p->next;
            delete p;
            p = head;
        } else {
            prev->next = p->next;
            delete p;
            p = prev->next;
        }
        counter++;
    } else {
        prev = p;
        p = p->next;
```

```
        }
    }

    cout << "Deleted " << counter << " elements\n";
    cout << "Remaining elements:\n";

    p = head;
    while (p != nullptr) {
        cout << p->word << endl;
        const q_elem* temp = p;
        p = p->next;
        delete temp;
    }

    return 0;
}
```

Тестовые данные и результаты тестирования: тестовые данные и резкльтаты тестирования изображены на рисунке 2

Ввод	Вывод
5 apple banana cherry apricot pear	Deleted 4 elements Remaining elements: cherry
1 Apple	Deleted 1 elements Remaining elements:

Рисунок 2 — Тестовые данные и результаты тестирования

Вывод: Программа реализует очередь, успешно удаляет слова, содержащие букву 'a', подсчитывает количество удалений и корректно освобождает динамическую память. Тестирование показало, что код работает верно для всех проверенных случаев.

Задача 2. Файлы

Задание: Создать текстовый файл F. Переписать из файла F в файл G все четные строки.

Выполнение: схема алгоритма изображена на рисунке 3. Текст программы изображен на рисунке 4.

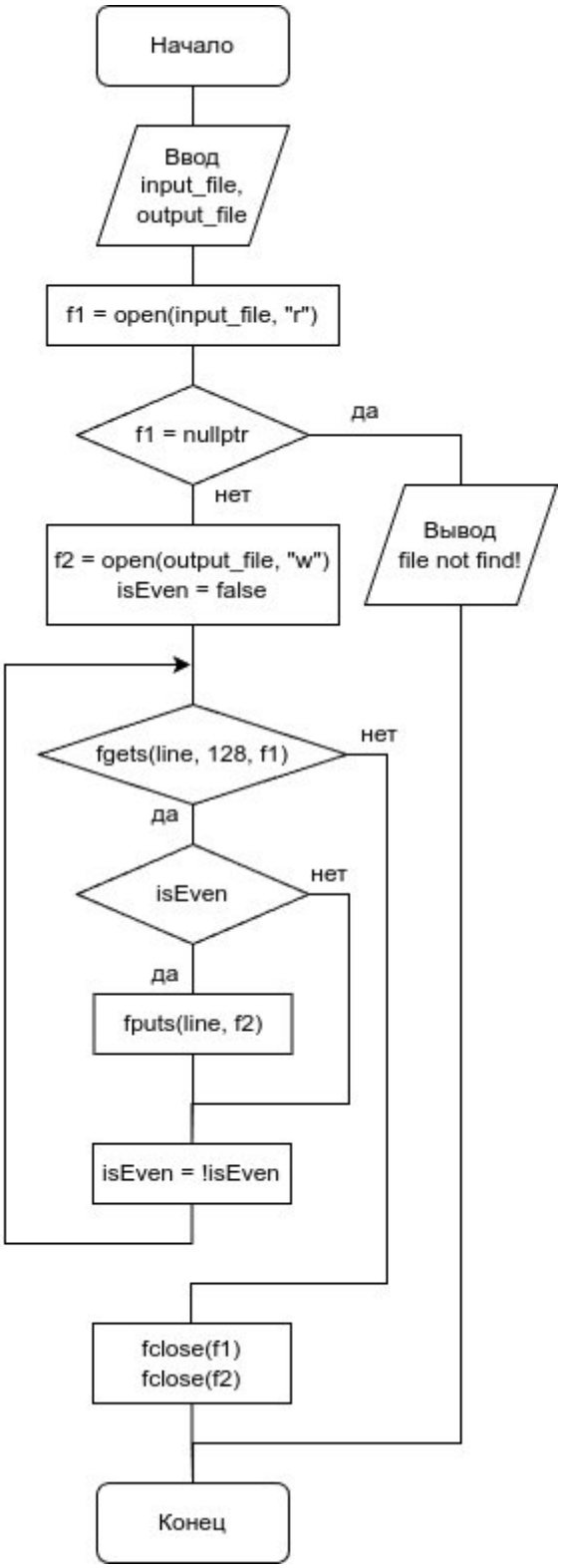

Рисунок 3 — Схема алгоритма

```cpp
#include <iostream>
using namespace std;

int main() {
    char input_file[128], output_file[128];
    cout << "Enter input filename: ";
    cin >> input_file;
    cout << "Enter output filename: ";
    cin >> output_file;

    FILE *f1 = fopen(filename:input_file, modes:"r");
    if (f1 == nullptr) {
        cout << "File not found!\n";
        return -1;
    }
    FILE *f2 = fopen(filename:output_file, modes:"w");
    char line[128];
    bool isEven = false;
    while (fgets(line, n:128, f1)) {
        if (isEven)
            fputs(line, f2);
        isEven = !isEven;
    }

    fclose(f1);
    fclose(f2);

    return 0;
}
```

Рисунок 4 — Код программы

Тестовые данные и результаты тестирования: тестовые данные и результаты тестирования изображены на рисунке 5

Ввод (input.txt)	Вывод (output.txt)
one two three four five six seven	two four six

Рисунок 5 — Тестовые данные и результаты тестирования

Вывод: Программа была успешно реализована. Тестирование показало, что работа программы корректна для всех проверенных случаев.