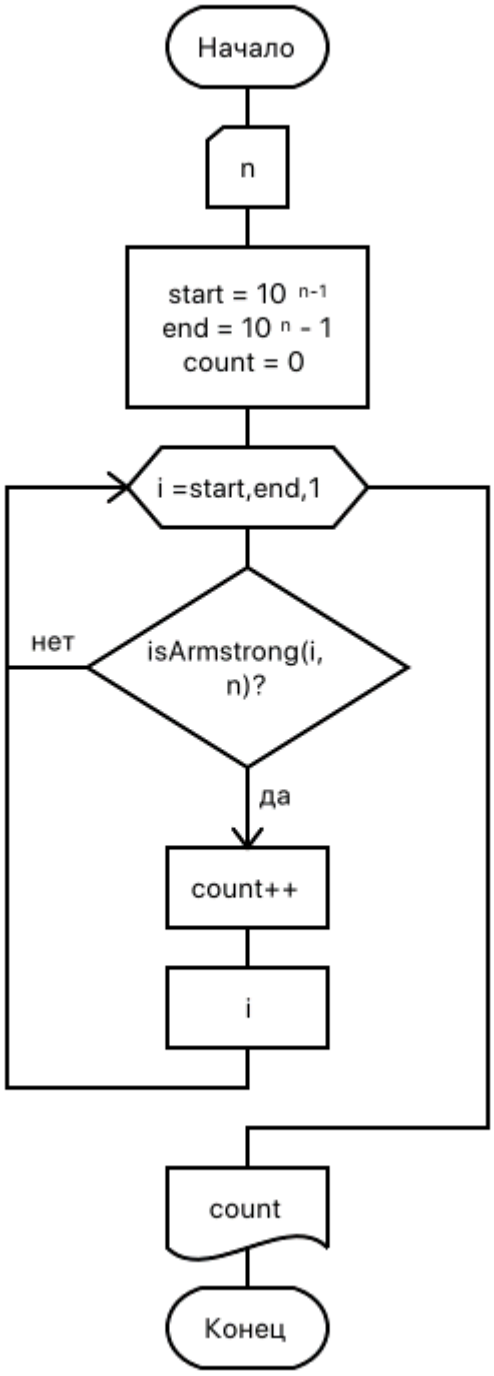
Задание 2. Программирование на C++

Часть 2.1. Функции

Число из n цифр является числом Армстронга, если сумма цифр этого числа, возведенных в степень n, равна самому этому числу. Например: 153 = 1*1*1 + 5*5*5 + 3*3*3. Написать программу, определяющую все числа Армстронга, состоящие из n цифр (n вводится с клавиатуры). Вывести на печать сами числа их количество. При программировании использовать функции.

Код решения

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

using namespace std;

bool isArmstrong(unsigned int const obj, unsigned int n) {
    auto tmp = obj;
    int sum = 0;
    for (int i = 0; i < n; ++i) {
        sum += pow(tmp % 10, n);
        tmp /= 10;
    }
    return sum == obj;
}

int main() {
    unsigned int n;
    cout << "Enter N: ";
    cin >> n;

    unsigned int start = pow(10, n - 1);
    auto end = (unsigned int) pow(10, n) - 1;

    auto count = 0;
    for (auto i = start; i <= end; ++i)
        if (isArmstrong(i, n)) {
            cout << i << endl;
            count++;
        }

    cout << "Total count: " << count << endl;
}
```



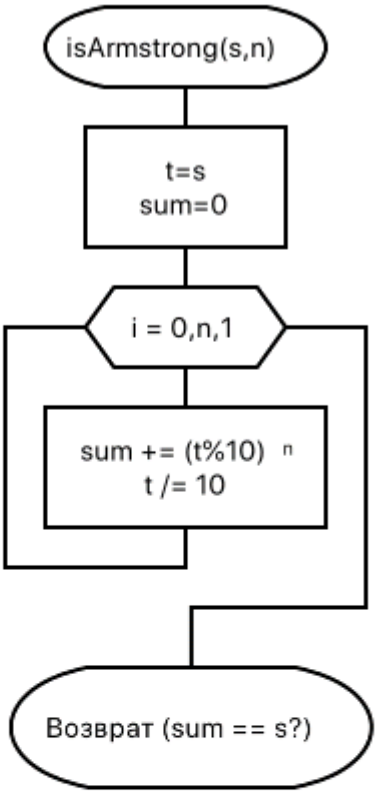


Figure 1. Схема алгоритма

Результаты работы

```
Enter N: 1
1
2
3
4
5
6
7
8
9
Total count: 9

Enter N: 3
153
370
371
407
Total count: 4
```

Часть 2.2. Текстовая обработка

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в удалении или замене слов. Если слово стоит на четном месте и начинается на букву «e», то оно удаляется; если слово стоит на четном месте и начинается на букву «c», то оно замещается на слово, введенное с клавиатуры. Вывести на печать исходную и скорректированную последовательности строк.

Код решения

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

int main() {
    vector<string> inputv, resultv;
    cout << "Enter string or leave empty:" << endl;
    while (true) {
        string tmp;
        getline(cin, tmp);
        if (tmp.empty())
            break;
        inputv.push_back(tmp);
        resultv.emplace_back("");
    }

    for (int r = 0; r < inputv.size(); r++) {
        auto &input = inputv[r];
        auto &result = resultv[r];
        auto len = input.length();

        int words = 0;
        bool nextIsWord = false;
        for (int i = 0; i < len; i++) {

            if (nextIsWord && words % 2 == 1) {
                switch (input[i]) {
                    case 'c': {
                        string tmp;
```

```
                        cout << "Enter replacement:" << endl;
                        cin >> tmp;
                        result.append(tmp);
                        // skip until next space
                        while (i + 1 < len && input[i + 1] != ' ')
                            i++;
                        break;
                    }
                    case 'e':
                        // skip after next word
                        while (i < len && input[i] != ' ')
                            i++;
                        break;
                    default:
                        result.push_back(input[i]);
                }
            } else {
                result.push_back(input[i]);
            }

            if (input[i] == ' ') {
                words++;
                nextIsWord = true;
            } else {
                nextIsWord = false;
            }
//            words += (nextIsWord = (input[i] == ' '));
        }
    }

    cout << endl << "Source lines:" << endl;
    for (const auto &str : inputv)
        cout << str << endl;

    cout << endl << "Result lines:" << endl;
    for (const auto &str : resultv)
        cout << str << endl;

    return 0;
}
```

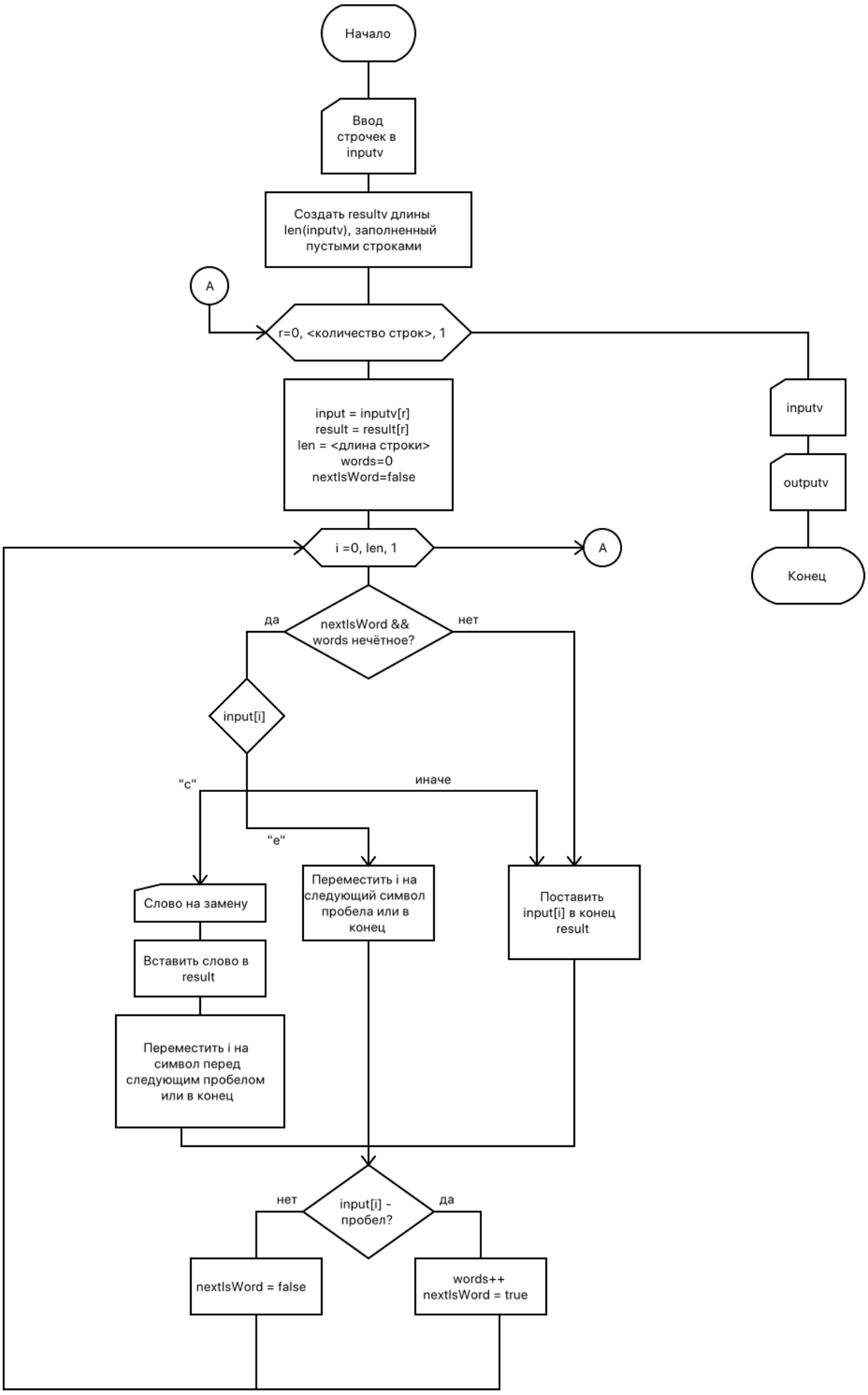

Начало
Ввод строчек в inputv
Создать resultv длины len(inputv), заполненный пустыми строками
A
r=0, <количество строк>, 1
input = inputv[r]
result = result[r]
len = <длина строки>
words=0
nextIsWord=false
inputv
outputv
Конец
i =0, len, 1
A
да
nextIsWord && words нечётное?
нет
input[i]
"c"
иначе
"e"
Слово на замену
Вставить слово в result
Переместить i на символ перед следующим пробелом или в конец
Переместить i на следующий символ пробела или в конец
Поставить input[i] в конец result
нет
input[i] - пробел?
да
nextIsWord = false
words++
nextIsWord = true

Figure 2. Схема алгоритма

Table 1. Результаты работы

```
Enter string or leave empty:
leave leave leave creplace
leave erase leave creplace

Enter replacement:
first
Enter replacement:
second

Source lines:
leave leave leave creplace
leave erase leave creplace

Result lines:
leave leave leave first
leave leave second
```

Часть 2.3. Файлы

Организовать программным способом файл F, состоящий из символьных строк. Переписать в файл G строки файла F, записав символы этих строк в обратном порядке и удалив символы e,E,r,t,U. При возникновении непредвиденных ситуаций выдать соответствующие сообщения. Вывести на экран оба файла.

Код решения

```cpp
#include <iostream>
#include <fstream>

using namespace std;

int main() {
    ifstream F;
    F.open("./F");

    fstream G;
    G.open("G", fstream::out);

    if (!F.is_open() || !G.is_open()) return -1;

    string line;
    while (getline(F, line)) {
        for (auto i = line.length() - 1; i >= 0 && i < line.length(); i--) {
            auto c = line[i];
            switch (c) {
                case 'e':
                case 'E':
                case 'r':
                case 't':
                case 'U':
                    break;
                default:
                    G << c;
            }
        }
        G << endl;
    }
    G.close();
    F.close();

    G.open("G", fstream::in);
    F.open("F", fstream::in);

    cout << "[[F file]]" << endl;
    cout << F.rdbuf() << endl;

    cout << "[[G file]]" << endl;
    cout << G.rdbuf() << endl;

    F.close();
    G.close();

    return 0;
}
```

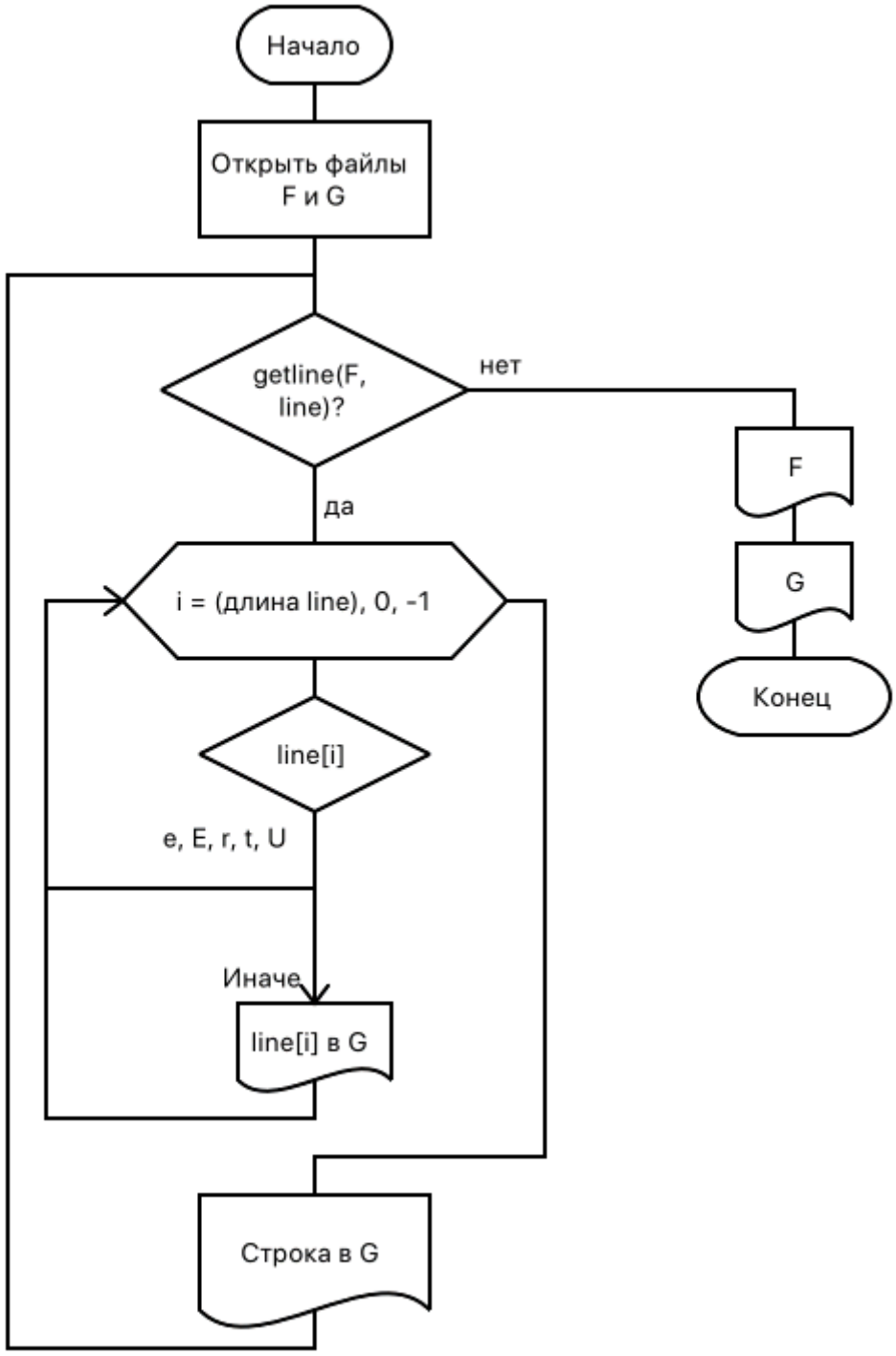


Figure 3. Схема алгоритма

Результат работы

```
[[F file]]
tester
12345 tEs7
Undefined behaviour
[[G file]]
s
7s 54321
uoivahb dnifdn
```