

Министерство науки и высшего образования Российской
Федерации Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Институт цифрового развития Кафедра инфокоммуникаций
ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1
дисциплины «Алгоритмизация» Вариант ____

Выполнил: Дзугев Альберт Мухаметович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и вычислительная
техника», направленность (профиль)
«Программное обеспечение средств вычислительной
техники и автоматизированных систем»,
очная форма обучения
_____ (подпись)

Руководитель практики: Воронкин Р.А., канд. техн. наук, доцент
кафедры инфокоммуникаций _____
(подпись)

Отчет защищен с оценкой _____ Дата
защиты _____ Ставрополь, 2023 г.

Порядок выполнения работы:

1. Выполнение задания номер 16 из ЕГЭ по информатике:

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(n) = F(n-1) * (2*n + 1), \text{ при } n > 1$$

Чему равно значение функции $F(4)$? В ответе запишите только натуральное число.

Рисунок 1. Задание номер 16.

```
1
2     #include <iostream>
3
4     using namespace std;
5
6     int F(int n) {
7         if (n == 1) {
8             return 1;
9         }
10        else {
11            return F(n - 1) * (2 * n + 1);
12        }
13    }
14
15    int main() {
16        int n;
17        cout << "CIN n: ";
18        cin >> n;
19        cout << "F(" << n << ") = " << F(n) << endl;
20        return 0;
21    }
22
```

Рисунок 2. Программа для выполнения 16 задания.

```
CIN n: 4
F(4) = 315

C:\Users\dzuev\OneDrive\Рабочий стол\Задание 16\x64\Debug\Задание 16.exe (процесс 5136) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 3. Ответ на 16 задание.

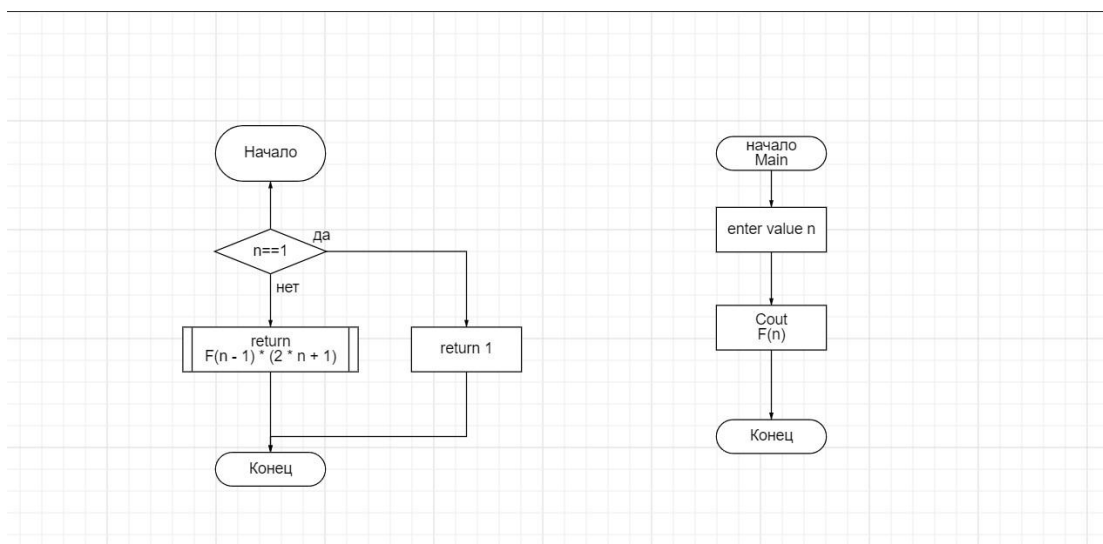


Рисунок 4. Блок-схема к 16 заданию.

2. Выполнение задния 17 из ЕГЭ по информатике.

[Версия для печати и копирования в MS Word](#)

Тип 17 № 59784

В файле содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от -1000000 до 1000000 включительно.

Задание 17

Определите количество троек элементов в которых только одно число четырехзначное, и сумма элементов тройки меньше максимального элемента последовательности оканчивающегося на 15. В ответе запишите два числа: сначала количество найденных троек, а затем минимальную из сумм таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Рисунок 5. Задание 17.

```

#include <iostream>
#include <fstream>
#include <vector>
#include <math.h>
using namespace std;
int main()
{
    std::ifstream inputFile("C:/Users/dzuev/OneDrive/Рабочий стол/python1/Программа проверяющая лежит ли точка на отрезке/17.txt");
    std::vector<int> numbers;
    int number;
    int count = 0;

    while (inputFile >> number)
    {
        numbers.push_back(number);
    }
    inputFile.close();
    int max = 0;

    for (int i = 0; i < (int)numbers.size(); i++)
    {
        if (numbers[i] % 100 == 15 && numbers[i] > max)
            max = numbers[i];
    }
    int min_sum = max;

    for (int i = 0; i <= (int)numbers.size() - 2; i++)
    {
        int c = 0, sum = 0;
        int d[] = { numbers[i], numbers[i + 1], numbers[i + 2] };

        for (int j = 0; j < 3; j++)
        {
            sum += d[j];
            if (abs(d[j]) > 999 && abs(d[j]) < 10000)
                c++;
        }
        if (c == 1 && sum < max)
        {
            count++;
            if (sum < min_sum)
                min_sum = sum;
        }
    }
    cout << count << "    " << min_sum << endl;
}

```

Рисунок 6. Программа для выполнения 17 задания.

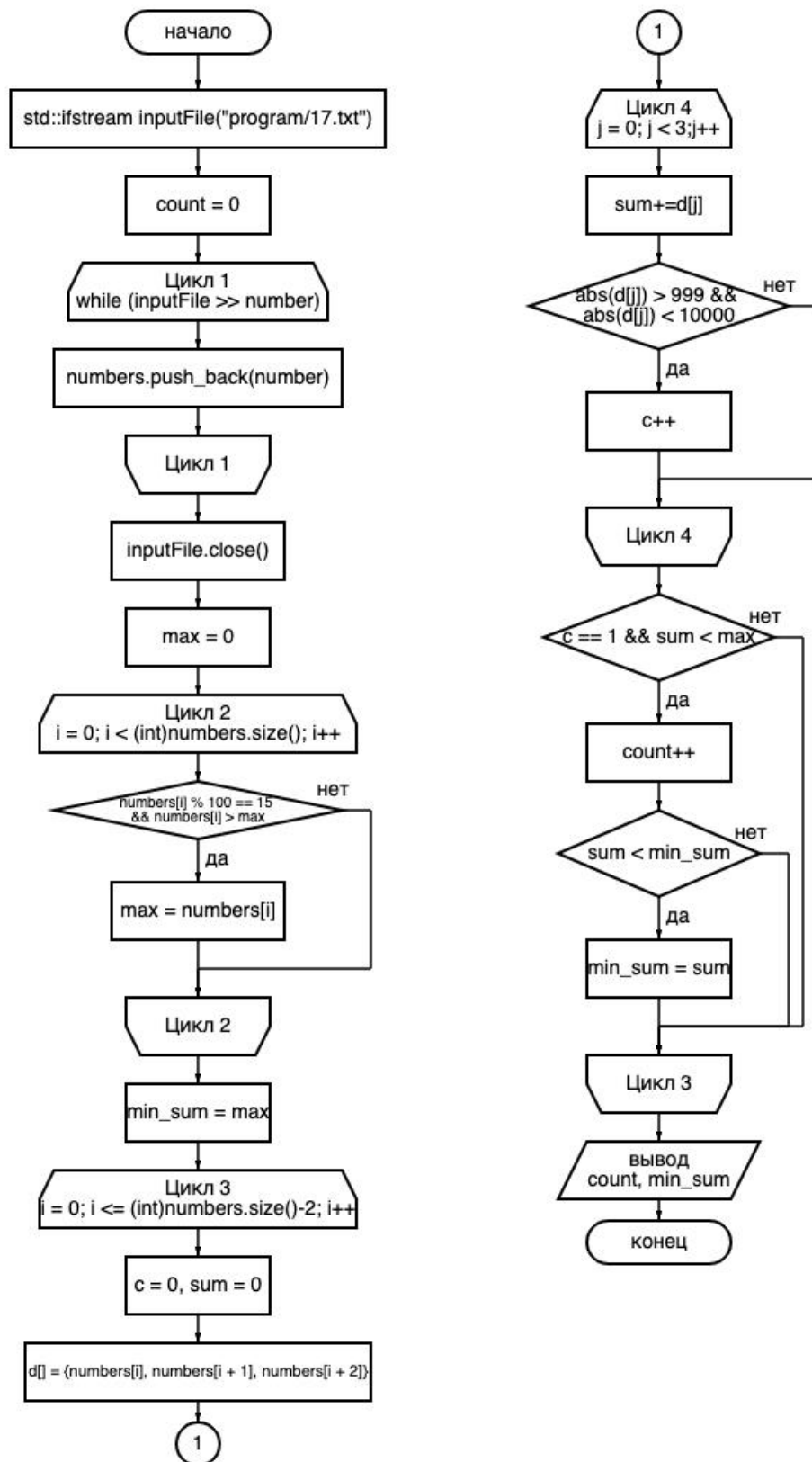


Рисунок 7. Блок-схема к 17 заданию.

Тип 18 № 36873

Дан квадрат 15×15 клеток, в каждой клетке которого записано целое число. В левом нижнем углу квадрата стоит робот. За один ход робот может переместиться на одну клетку вправо или на одну клетку вверх. Выходить за пределы квадрата робот не может. При этом ведётся подсчёт суммы по следующим правилам: число в очередной клетке, через которую проходит робот, включается в сумму, если оно больше числа в предыдущей клетке на пути робота. Если число в очередной клетке не больше числа в предыдущей, сумма не изменяется. Число в начальной клетке всегда включается в сумму. Необходимо переместить робота в правый верхний угол так, чтобы полученная сумма была максимальной. В ответе запишите максимально возможную сумму.

Исходные данные записаны в электронной таблице.

Задание 18

Пример входных данных (для таблицы размером 4×4):

44	42	89	37
18	35	50	20
6	41	26	64
7	9	70	85

Рисунок 8. Задание 18.

```
#include <iomanip>
#include <algorithm>
using namespace std;
template <typename T, size_t N, size_t M>
void printMatrix(T(&matrix)[N][M])
{
    for (int i = 0; i < 15; i++)
    {
        for (int j = 0; j < 15; j++)
        {
            std::cout << std::setw(4) << matrix[i][j] << " ";
        }
        std::cout << std::endl;
    }
}

int main()
{
    int matrix[15][15] = {
        {12, 88, 57, 63, 77, 70, 6, 98, 65, 58, 19, 73, 13, 86, 1},
        {12, 66, 3, 14, 36, 65, 51, 4, 45, 78, 12, 31, 91, 12, 86},
        {61, 41, 46, 70, 39, 7, 76, 23, 3, 4, 62, 75, 63, 65, 54},
        {86, 73, 65, 44, 42, 14, 50, 92, 94, 37, 6, 78, 46, 97, 17},
        {95, 40, 51, 62, 12, 14, 58, 49, 83, 97, 70, 14, 6, 47, 55},
        {75, 43, 22, 77, 71, 53, 19, 58, 37, 69, 48, 98, 98, 89, 35},
        {22, 84, 34, 52, 37, 39, 72, 84, 19, 49, 10, 78, 75, 7, 42},
        {59, 93, 9, 90, 13, 31, 93, 54, 73, 52, 4, 94, 37, 81, 22},
        {68, 76, 84, 83, 87, 36, 26, 19, 19, 73, 24, 47, 42, 52, 5},
        {79, 12, 76, 34, 77, 2, 70, 20, 78, 49, 7, 74, 69, 90, 45},
        {90, 76, 32, 31, 46, 59, 35, 9, 10, 32, 93, 6, 19, 40, 60},
        {67, 21, 74, 59, 98, 15, 49, 70, 31, 48, 63, 52, 70, 33, 69},
        {46, 39, 51, 77, 78, 36, 46, 50, 12, 56, 75, 77, 36, 5, 11},
        {26, 62, 85, 7, 15, 74, 39, 94, 53, 89, 53, 55, 3, 49, 4},
        {21, 94, 40, 22, 35, 50, 85, 61, 29, 89, 18, 62, 52, 94, 16} };
    int newmatrix[15][15];
    newmatrix[14][0] = 21;
    for (int i = 13; i >= 0; i--)
    {
        newmatrix[i][0] = newmatrix[i + 1][0] + ((matrix[i][0] > matrix[i + 1][0]) ? matrix[i][0] : 0);
    }
    for (int j = 1; j <= 14; j++)
    {
        newmatrix[14][j] = newmatrix[14][j - 1] + ((matrix[14][j] > matrix[14][j - 1]) ? matrix[14][j] : 0);
    }
    for (int i = 13; i >= 0; i--)
    {
        for (int j = 1; j <= 14; j++)
        {
            int a = newmatrix[i + 1][j] + ((matrix[i][j] > matrix[i + 1][j]) ? matrix[i][j] : 0);
            int b = newmatrix[i][j - 1] + ((matrix[i][j] > matrix[i][j - 1]) ? matrix[i][j - 1] : 0);
            newmatrix[i][j] = max(a, b);
        }
    }
    printMatrix(newmatrix);
}
```

Рисунок 9. Код для решения задания 18.

420	647	647	734	844	902	902	1039	1173	1186	1205	1290	1317	1403	1403
420	559	584	671	767	832	899	941	1108	1186	1186	1217	1317	1317	1403
420	493	584	671	767	767	899	941	1063	1067	1129	1204	1226	1291	1315
420	493	584	601	767	767	823	941	1063	1066	1066	1152	1163	1261	1261
420	420	519	601	725	746	823	849	969	1066	1066	1074	1117	1164	1219
325	419	468	601	725	746	765	849	886	967	967	1074	1117	1124	1124
250	419	468	524	654	693	765	849	849	898	898	976	1019	1035	1077
250	419	434	524	617	648	741	741	814	814	814	944	944	1035	1035
250	326	434	434	617	617	617	617	678	751	800	850	864	954	954
250	250	350	350	530	530	600	600	678	678	776	850	864	954	954
250	250	274	277	453	512	512	572	582	620	776	776	795	835	895
160	160	274	277	453	453	502	572	572	620	683	683	753	753	822
93	115	200	277	355	355	401	451	451	507	582	659	659	659	670
47	115	200	200	215	289	289	383	383	472	472	527	527	576	576
21	115	115	115	150	200	285	285	285	374	374	436	436	530	530

Рисунок 10. Результат выполнения кода.

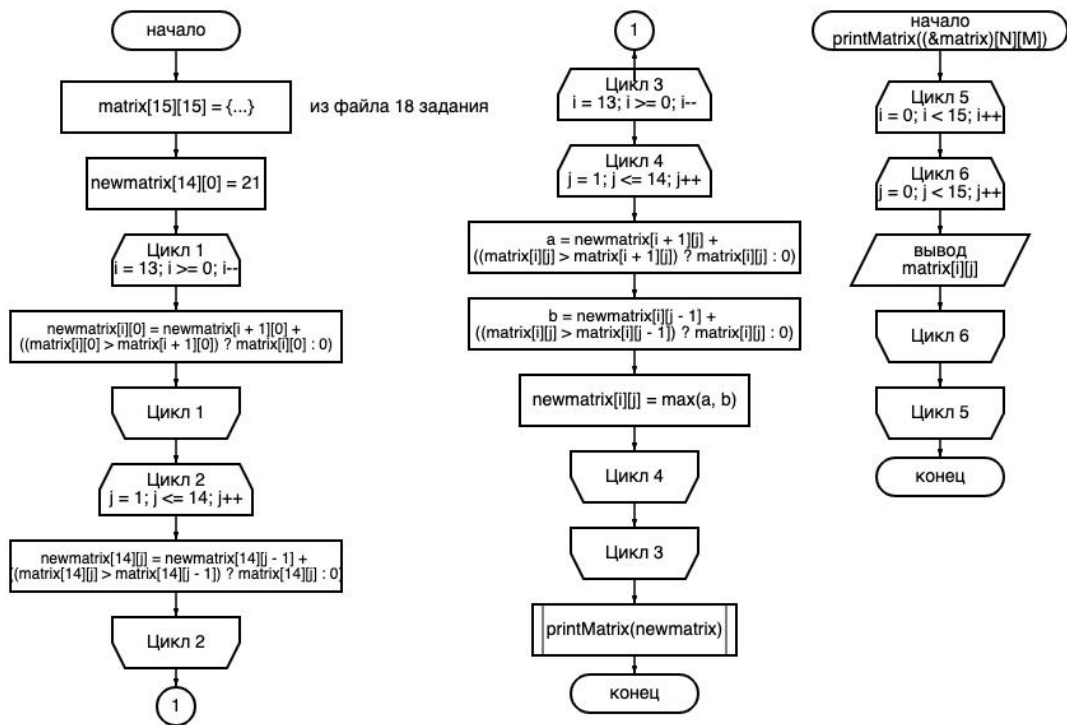


Рисунок 11. Блок-схема к 18 заданию

Вывод: Я повторил способы выполнения разных заданий