

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»**

Выполнила:  
Кубанова Ксения Олеговна  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А.

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

## Порядок выполнения работы

16.

Тип 16 № 4558

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

$$F(1) = 1$$

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

Чему равно значение функции  $F(5)$ ? В ответе запишите только натуральное число.

Рисунок 1: Задание 16

### Блок-схема

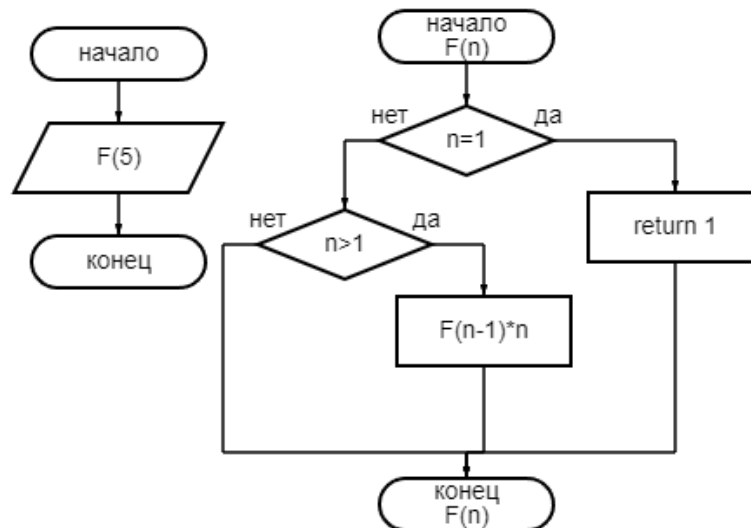


Рисунок 2: Задание 16, блок-схема

### Код

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

Рисунок 3: Задание 16, код

17.

Тип 17 № 39764

Файл содержит последовательность неотрицательных целых чисел, не превышающих 10 000. Назовём тройкой три идущих подряд элемента последовательности. Определите количество троек чисел таких, которые могут являться сторонами прямоугольного треугольника. В ответе запишите два числа: сначала количество найденных троек, а затем — максимальную сумму элементов таких троек. Если таких троек не найдётся — следует вывести 0 0.

Рисунок 4: Задание 17

## Блок-схема

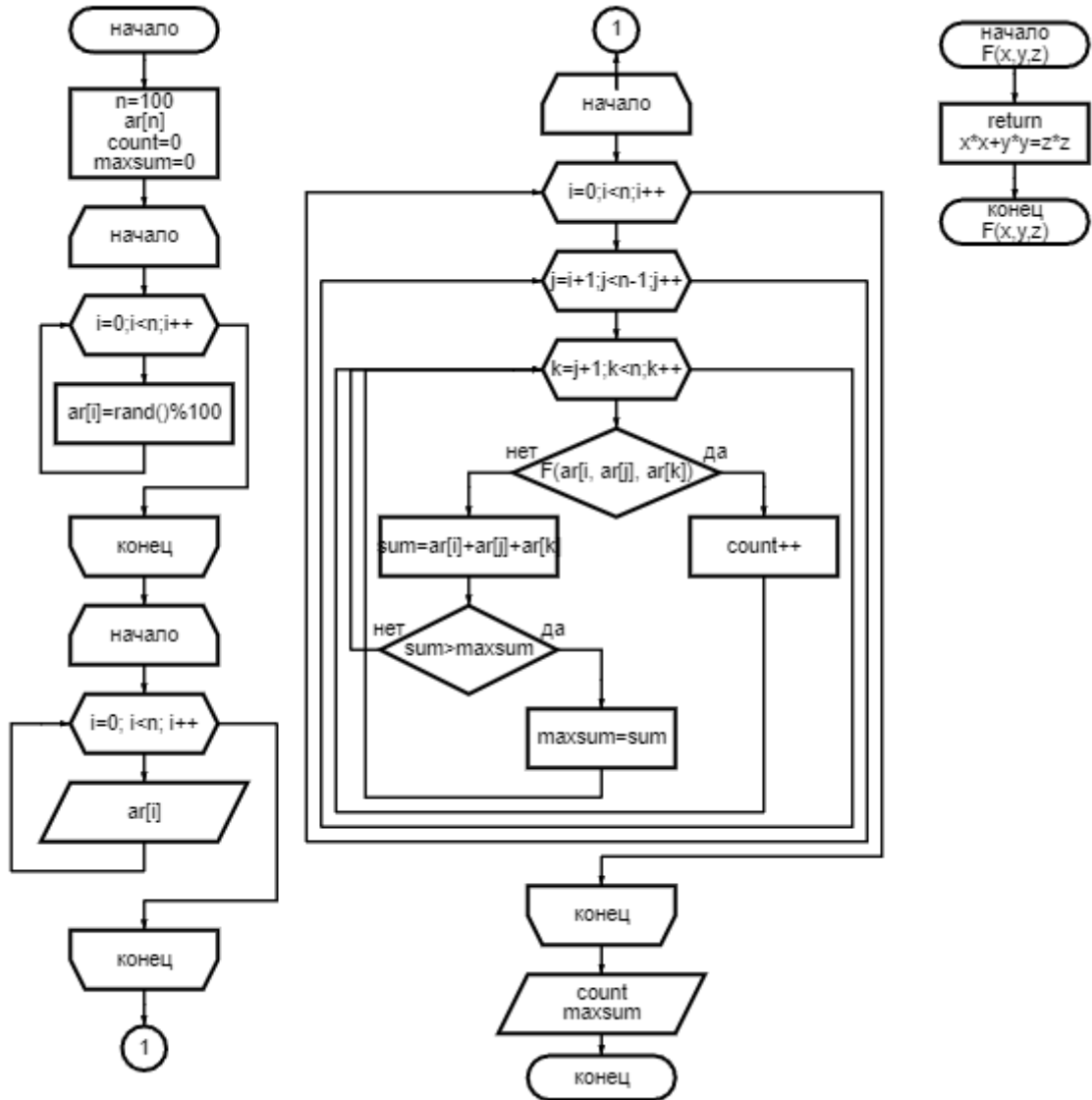


Рисунок 5: Задание 17, блок-схема

## Код

```

int f(int x, int y, int z)
{
    return x * x + y * y == z * z;
}

int main()
{
    setlocale(LC_ALL, "");
    const int n = 100;
    int ar[n];
    int count = 0;
    int maxsum = 0;
    cout << "Полученный массив:" << endl;
    for (int i = 0; i < n; i++)
        ar[i] = rand() % 100;
    for (int i = 0; i < n; i++)
        cout << ar[i] << " ";
    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n - 1; j++)
            for (int k = j + 1; k < n; k++)
            {
                if (f(ar[i], ar[j], ar[k]))
                    count++;
                int sum = ar[i] + ar[j] + ar[k];
                if (sum > maxsum)
                    maxsum = sum;
            }
    cout << endl << "Кол-во троек - " << count << endl;
    cout << "Максимальная сумма кол-ва троек - " << maxsum << endl;
}

```

Рисунок 6: Задание 17, код

18.

3 Тип 18 № 27667

Квадрат разлинован на  $N \times N$  клеток ( $1 < N < 17$ ). Исполнитель Робот может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: вправо или вниз. По команде вправо Робот перемещается в соседнюю правую клетку, по команде вниз — в соседнюю нижнюю. При попытке выхода за границу квадрата Робот разрушается. Перед каждым запуском Робота в каждой клетке квадрата лежит монета достоинством от 1 до 100. Посетив клетку, Робот забирает монету с собой; это также относится к начальной и конечной клетке маршрута Робота.

#### Задание 18

Откройте файл. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответ запишите два числа друг за другом без разделительных знаков — сначала максимальную сумму, затем минимальную.

Исходные данные представляют собой электронную таблицу размером  $N \times N$ , каждая ячейка которой соответствует клетке квадрата.

Рисунок 7: Задание 18

## Блок-схема

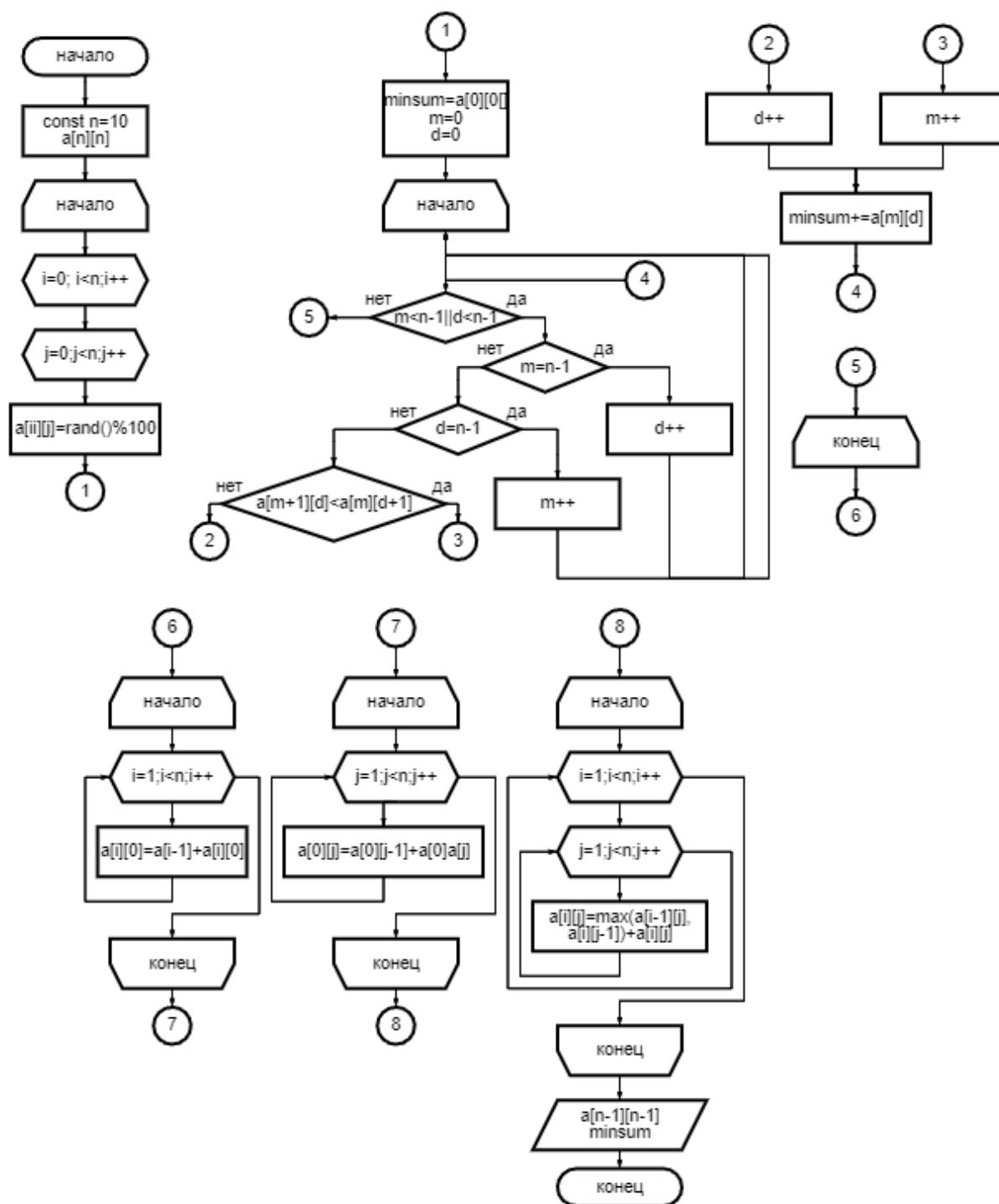


Рисунок 8: Задание 18, блок-схема

## Код

Ввод и вывод массива

```

const int n = 10;
int a[n][n];
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        a[i][j] = rand() % 100;
        cout << a[i][j] << '\t';
    }
}

```

Рисунок 9: Задание 18, код, часть 1

## Нахождение минимальной суммы

```

int minSum = a[0][0];
int m = 0;
int d = 0;

while (m < n - 1 || d < n - 1)
{
    if (m == n - 1)
    {
        d++;
    }
    else if (d == n - 1)
    {
        m++;
    }
    else
    {
        if (a[m + 1][d] < a[m][d + 1])
        {
            m++;
        }
        else
        {
            d++;
        }
    }
    minSum += a[m][d];
}

```

Рисунок 10: Задание 18, код, часть 2

## Нахождение максимальной суммы и вывод на экран решения

```

for (int i = 1; i < n; ++i)
{
    a[i][0] = a[i - 1][0] + a[i][0];
}
for (int j = 1; j < n; ++j)
{
    a[0][j] = a[0][j - 1] + a[0][j];
}
for (int i = 1; i < n; ++i)
{
    for (int j = 1; j < n; ++j)
    {
        a[i][j] = max(a[i - 1][j], a[i][j - 1]) + a[i][j];
    }
}
cout << "Max: " << a[n - 1][n - 1] << endl;
cout << "Min: " << minSum << endl;

```

Рисунок 11: Задание 18, код, часть 3