

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»**

Выполнил:  
Кожуховский Виктор Андреевич  
1 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 Роман Александрович

---

(подпись)

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

Ставрополь, 2023 г.

## Порядок выполнения работы:

### Выполнил задание 16:

#### 1 Тип 16 № 5057

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

$$F(1) = 3; F(2) = 3;$$

$$F(n) = 5 * F(n-1) - 4 * F(n-2) \text{ при } n > 2.$$

Чему равно значение функции  $F(15)$ ? В ответе запишите только натуральное число.

Ответ:

Рисунок 1. Условие задания 16

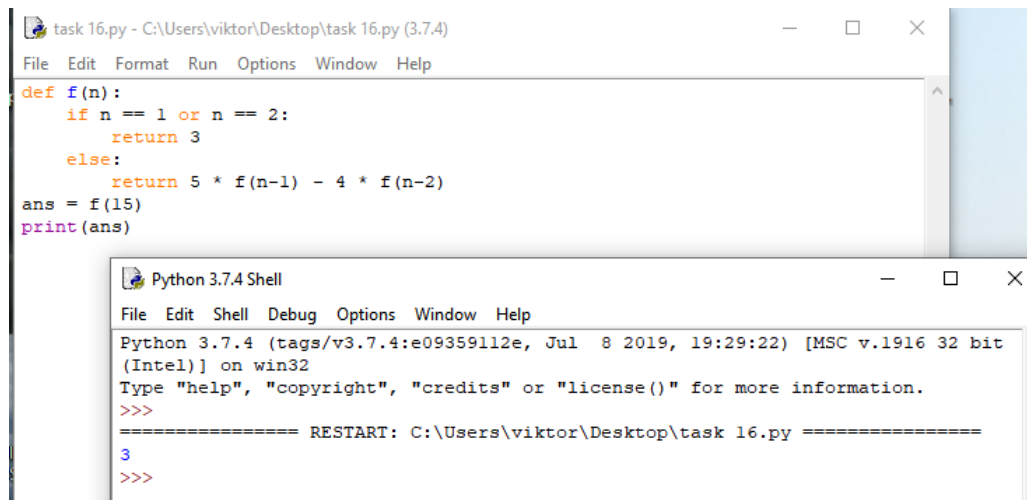


Рисунок 2. Код для решения задания 16

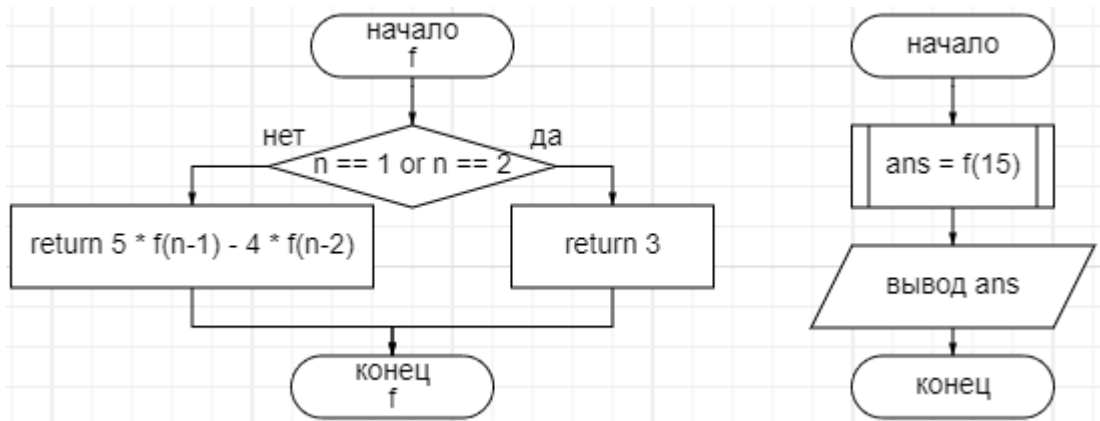


Рисунок 3. Блок-схема решения задания 16

### Выполнил задание 17:

#### 2 Тип 17 № 37368

В файле содержится последовательность из 10 000 целых положительных чисел. Каждое число не превышает 10 000. Определите и запишите в ответе сначала количество пар элементов последовательности, у которых сумма элементов кратна 60 и хотя бы один элемент из пары делится на 40, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два различных элемента последовательности. Порядок элементов в паре не важен.

[17.txt](#)

Ответ:

Рисунок 4. Условие задания 17

```

task 17.py - C:\Users\viktor\Desktop\task 17.py (3.7.4)
File Edit Format Run Options Window Help

cnt = maxS = 0
file = open('17.txt')
a = [int(i) for i in file]
for i in range(len(a)-1):
    for j in range(i+1, len(a)):
        if (a[i] + a[j]) % 60 == 0 and (a[i] % 40 == 0 or a[j] % 40 == 0):
            cnt += 1
            maxS = max(maxS, a[i] + a[j])
print(cnt, maxS)

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MS
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more info
>>>
===== RESTART: C:\Users\viktor\Desktop\task 17.py =====
29278 19860
>>> |
  
```

Рисунок 5. Код для решения задания 17

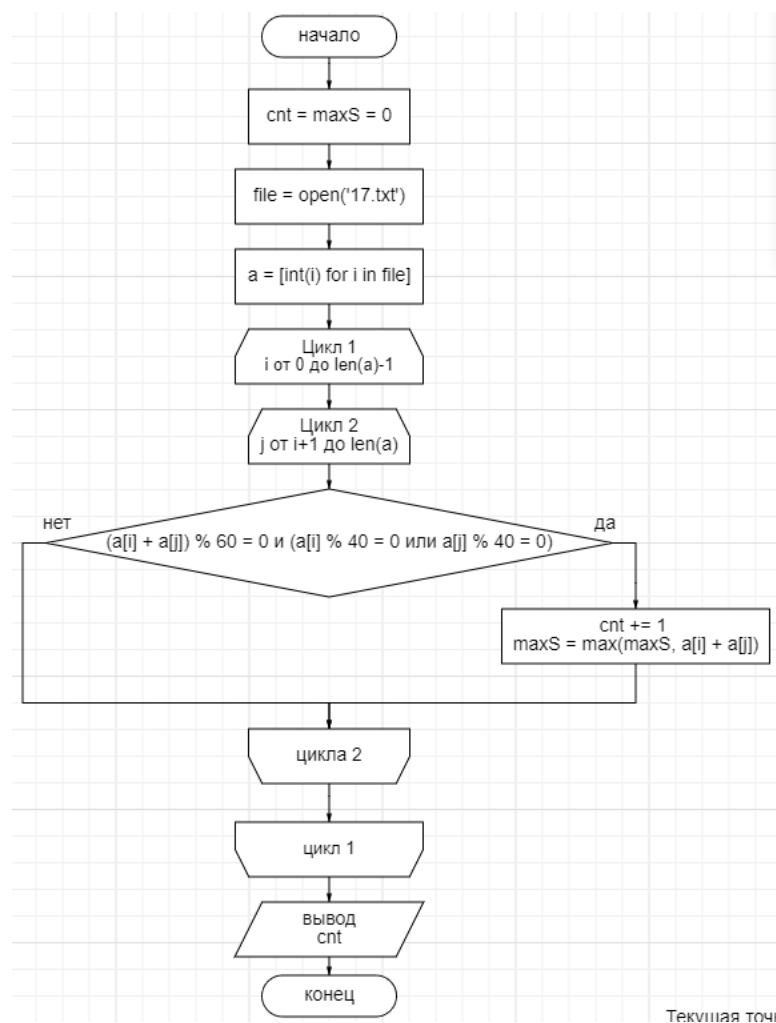


Рисунок 6. Блок-схема решения задания 17

## Выполнил задание 18:

### 1 Тип 18 № 27679

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.

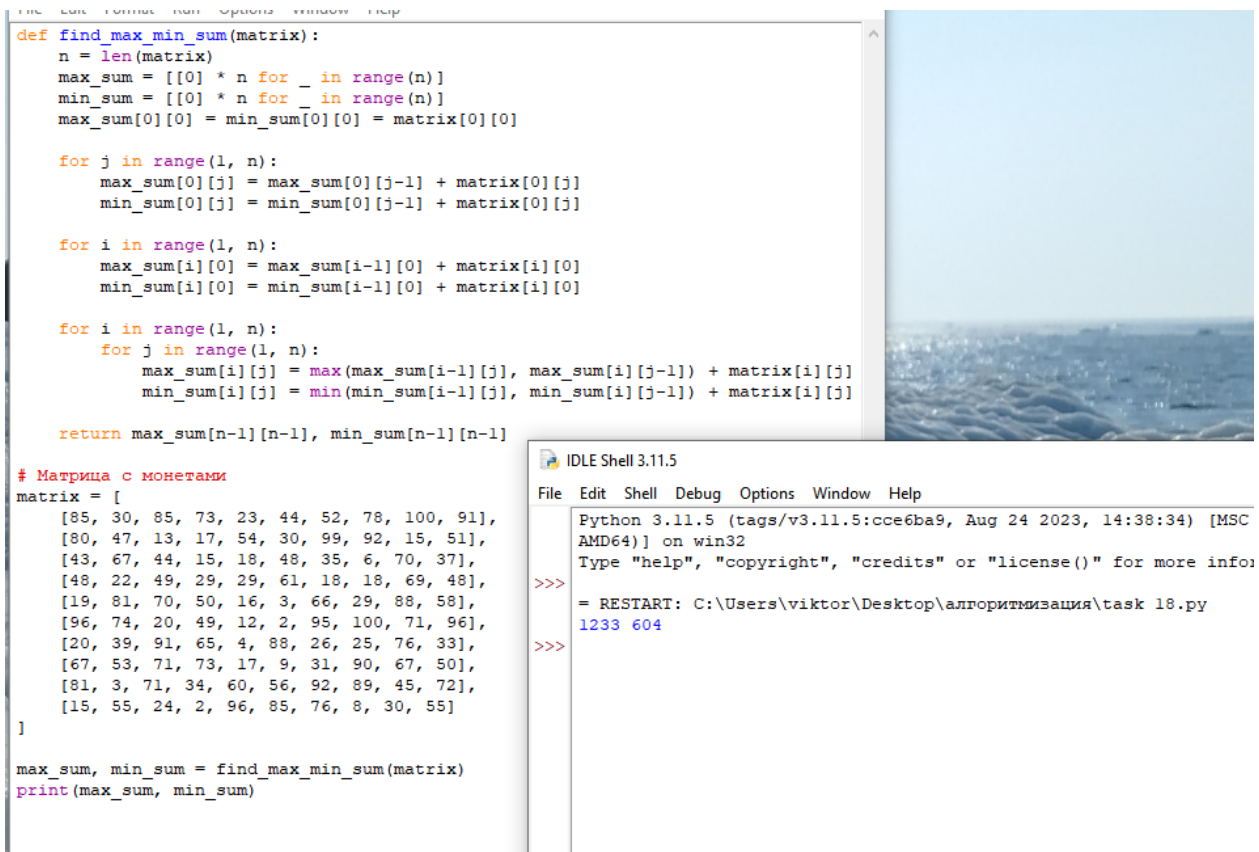
Пример входных данных:

1	8	8	4
10	1	1	3
1	3	12	2
2	3	5	6

Для указанных входных данных ответом должна быть пара чисел 41 и 22.

Ответ:

## Рисунок 7. Условие задания 18



```
def find_max_min_sum(matrix):
    n = len(matrix)
    max_sum = [[0] * n for _ in range(n)]
    min_sum = [[0] * n for _ in range(n)]
    max_sum[0][0] = min_sum[0][0] = matrix[0][0]

    for j in range(1, n):
        max_sum[0][j] = max_sum[0][j-1] + matrix[0][j]
        min_sum[0][j] = min_sum[0][j-1] + matrix[0][j]

    for i in range(1, n):
        max_sum[i][0] = max_sum[i-1][0] + matrix[i][0]
        min_sum[i][0] = min_sum[i-1][0] + matrix[i][0]

    for i in range(1, n):
        for j in range(1, n):
            max_sum[i][j] = max(max_sum[i-1][j], max_sum[i][j-1]) + matrix[i][j]
            min_sum[i][j] = min(min_sum[i-1][j], min_sum[i][j-1]) + matrix[i][j]

    return max_sum[n-1][n-1], min_sum[n-1][n-1]

# Матрица с монетами
matrix = [
    [85, 30, 85, 73, 23, 44, 52, 78, 100, 91],
    [80, 47, 13, 17, 54, 30, 99, 92, 15, 51],
    [43, 67, 44, 15, 18, 48, 35, 6, 70, 37],
    [48, 22, 49, 29, 29, 61, 18, 18, 69, 48],
    [19, 81, 70, 50, 16, 3, 66, 29, 88, 58],
    [96, 74, 20, 49, 12, 2, 95, 100, 71, 96],
    [20, 39, 91, 65, 4, 88, 26, 25, 76, 33],
    [67, 53, 71, 73, 17, 9, 31, 90, 67, 50],
    [81, 3, 71, 34, 60, 56, 92, 89, 45, 72],
    [15, 55, 24, 2, 96, 85, 76, 8, 30, 55]
]

max_sum, min_sum = find_max_min_sum(matrix)
print(max_sum, min_sum)
```

## Рисунок 8. Код для решения задания 18

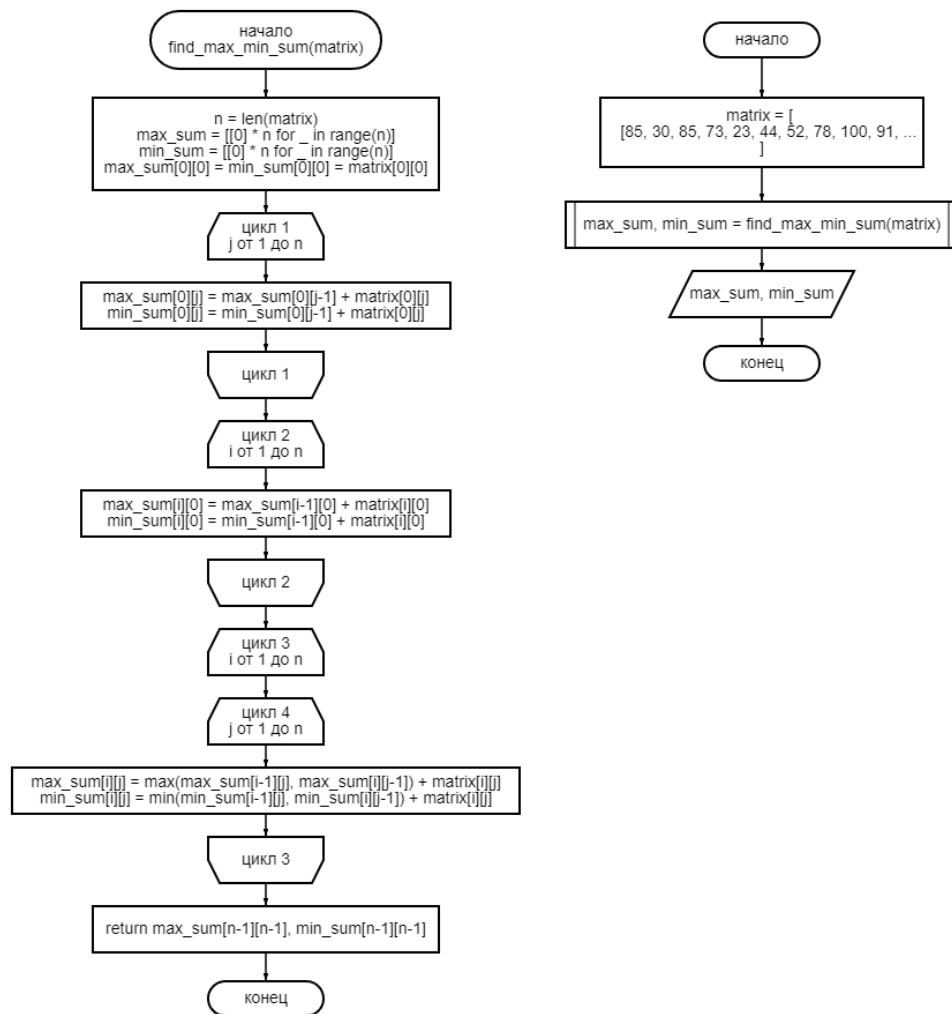


Рисунок 9. Блок-схема решения задания 18