

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 ЛАБОРАТОРНОЙ РАБОТЕ №1**  
**дисциплины**  
**«Алгоритмизация»**

Выполнил:  
Ибрагимов Муса Айнудинович  
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:  
Воронкин Роман Александрович

---

(подпись)

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

Ставрополь, 2023 г.

### Порядок выполнения работы:

**Задание 16.** Алгоритм вычисления значения функции  $F(n)$ , где  $n$  — целое неотрицательное число, задан следующими соотношениями:

$$F(0) = 0;$$

$$F(n) = F(n - 1) + 1, \text{ если } n \text{ нечётно};$$

$$F(n) = F(n / 2), \text{ если } n > 0 \text{ и при этом } n \text{ чётно}.$$

Укажите количество таких значений  $n < 1\,000\,000\,000$ , для которых  $F(n) = 3$ .

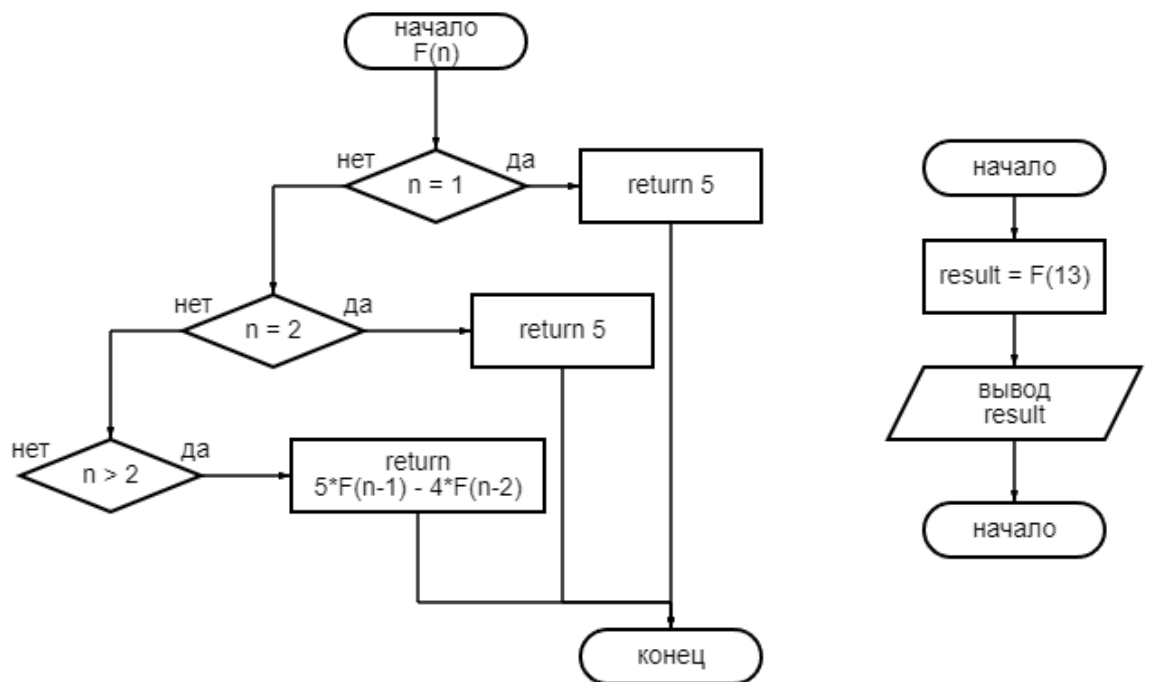
1) Написал программу для вычисления значения функции  $F(n)$ .

```
*223.py - C:/Users/User/AppData/Local/Programs/Python/Python311/223.py (3.11.0)*
File Edit Format Run Options Window Help
def F(n):
    if n == 1:
        return 5
    if n == 2:
        return 5
    if n > 2:
        return 5 * F(n-1) - 4 * F(n-2)
print(F(13))
```

2) Результат выполнения программы.

```
IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/User/AppData/Local/Programs/Python/Python311/223.py =====
5
>>> |
```

3) Составил блок-схему для программы.



**Задание 17.** Файл содержит последовательность неотрицательных целых чисел, не превышающих 10 000. Назовём парой два идущих подряд элемента последовательности. Определите количество пар, в которых хотя бы один из двух элементов делится на 3, а их сумма делится на 5. В ответе запишите два числа: сначала количество найденных пар, а затем – максимальную сумму элементов таких пар.

1) Написал программу на Python для задания 17.

223.py - C:/Users/User/AppData/Local/Programs/Python/Python311/223.py (3.11.0)

File Edit Format Run Options Window Help

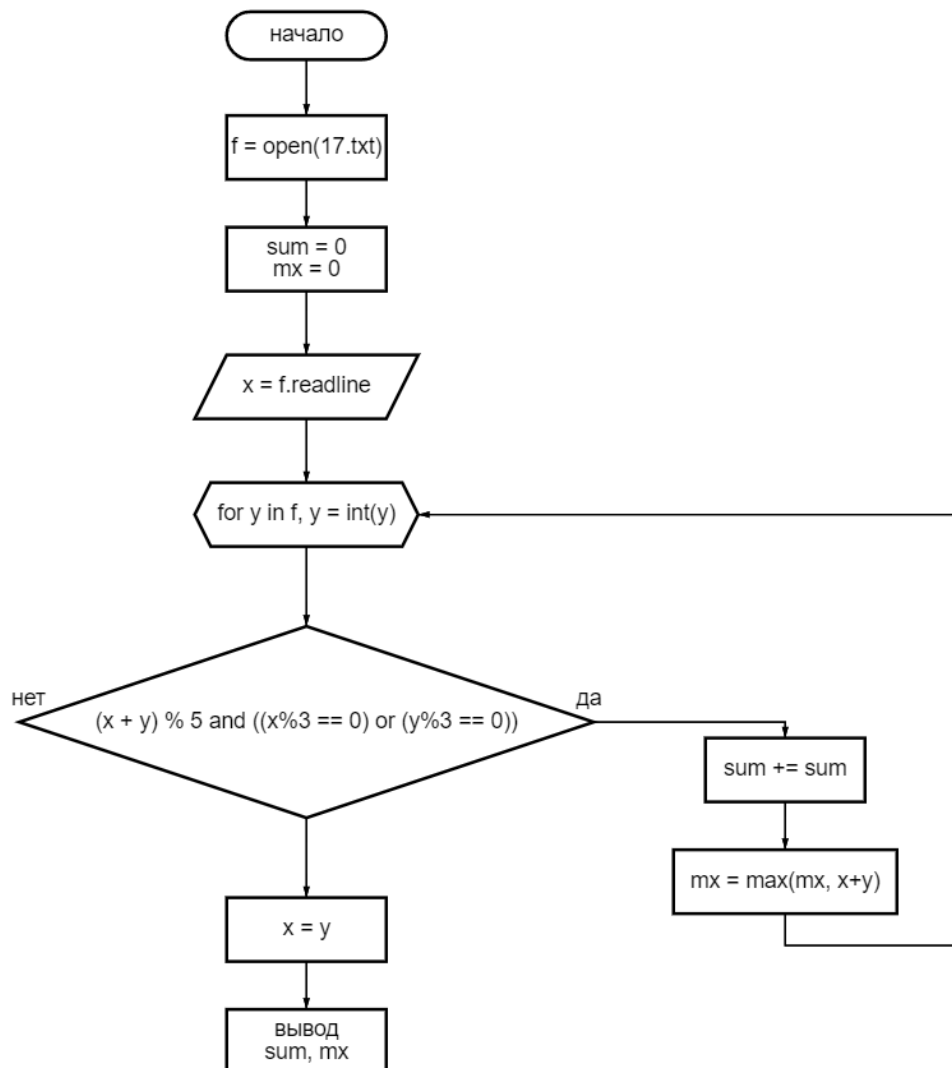
```

f=open('17_38951.txt')
Sum=0
Mx=0
x=int(f.readline())
for y in f:
    y=int(y)
    if (((x+y)%5==0) and ((x%3==0) or (y%3==0))):
        Sum += 1
        Mx = max(Mx, x+y)
    x=y
print(Sum, Mx)
  
```

## 2) Результат выполнения программы.

```
IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/User/AppData/Local/Programs/Python/Python311/223.py =====
635 19730
>>>
```

## 3) Написал блок-схему к программе.



**Задание 18.** 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 вверх. По команде вправо Робот перемещается в соседнюю правую клетку, по команде вверх — в соседнюю верхнюю. 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.

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

1) Написал программу, которая определяет максимальную денежную сумму, которую может собрать Робот, пройдя из левой нижней клетки таблицы в правую верхнюю.

```

1  // Алгоритмизация. Лабораторная работа №1.
2  // Ибрагимов Муса, ИБТ-6-о-22-1
3
4  #include <iostream>
5  using namespace std;
6  int maxElement;
7
8  int main() {
9      setlocale(LC_ALL, "ru");
10     int array[10][10] = {
11         {71, 81, 6, 50, 44, 16, 22, 10, 83, 1},
12         {20, 39, 80, 95, 60, 59, 95, 76, 17, 27},
13         {43, 7, 25, 70, 52, 36, 23, 47, 69, 35},
14         {66, 44, 55, 77, 18, 13, 75, 66, 98, 94},
15         {92, 44, 45, 54, 34, 45, 5, 16, 4, 40},
16         {2, 68, 34, 50, 65, 49, 86, 75, 91, 28},
17         {1, 28, 84, 18, 51, 23, 44, 79, 84, 36},
18         {30, 71, 77, 73, 15, 52, 94, 28, 82, 39},
19         {59, 52, 27, 25, 24, 56, 3, 87, 84, 46},
20         {67, 4, 79, 8, 39, 47, 31, 18, 65, 74},
21     };
22     for (int i = 8; i >= 0; i--) {
23         array[i][0] += array[i + 1][0];
24     }
25     for (int j = 1; j < 10; j++)
26     {
27         array[9][j] += array[9][j - 1];
28     }
29
30     for (int i = 8; i >= 0; i--) {
31         for (int j = 1; j < 10; j++) {
32             int maxElement = max(array[i][j - 1], array[i + 1][j]);
33             array[i][j] += maxElement;
34         }
35     }
36     for (int i = 0; i < 10; i++) {
37         for (int j = 0; j < 10; j++) {
38
39         }
40     }
41     cout << "Максимальная сумма = " << array[0][9] << endl;

```

## 2) Результат выполнения программы.

Максимальная сумма = 1181  
  
C:\Users\User\source\repos\ConsoleApplication2\Debug\ConsoleApplication2.exe (процесс)  
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Средство отладки".  
Нажмите любую клавишу, чтобы закрыть это окно...

## 3) Написал программу, которая определяет минимальную денежную сумму, которую может собрать Робот, пройдя из левой нижней клетки таблицы в правую верхнюю.

```
1  // Алгоритмизация. Лабораторная работа №1.  
2  // Ибрагимов Муса, ИБТ-6-о-22-1  
3  
4  #include <iostream>  
5  using namespace std;  
6  int maxElement;  
7  
8  int main() {  
9      setlocale(LC_ALL, "ru");  
10     int array[10][10] = {  
11         {71, 81, 6, 50, 44, 16, 22, 10, 83, 1},  
12         {20, 39, 80, 95, 60, 59, 95, 76, 17, 27},  
13         {43, 7, 25, 70, 52, 36, 23, 47, 69, 35},  
14         {66, 44, 55, 77, 18, 13, 75, 66, 98, 94},  
15         {92, 44, 45, 54, 34, 45, 5, 16, 4, 40},  
16         {2, 68, 34, 50, 65, 49, 86, 75, 91, 28},  
17         {1, 28, 84, 18, 51, 23, 44, 79, 84, 36},  
18         {30, 71, 77, 73, 15, 52, 94, 28, 82, 39},  
19         {59, 52, 27, 25, 24, 56, 3, 87, 84, 46},  
20         {67, 4, 79, 8, 39, 47, 31, 18, 65, 74},  
21     };  
22     for (int i = 8; i >= 0; i--) {  
23         array[i][0] += array[i + 1][0];  
24     }  
25     for (int j = 1; j < 10; j++)  
26     {  
27         array[9][j] += array[9][j - 1];  
28     }  
29  
30     for (int i = 8; i >= 0; i--) {  
31         for (int j = 1; j < 10; j++) {  
32             int maxElement = min(array[i][j - 1], array[i + 1][j]);  
33             array[i][j] += maxElement;  
34         }  
35     }  
36     for (int i = 0; i < 10; i++) {  
37         for (int j = 0; j < 10; j++) {  
38             }  
39         }  
40     }  
41     cout << "Минимальная сумма = " << array[0][9] << endl;
```

#### 4) Результат выполнения программы.

```
Консоль отладки Microsoft Visual Studio
Минимальная сумма = 604

C:\Users\User\source\repos\ConsoleApplication2\Debug\ConsoleApplication2.exe
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр
"Отладка -> Инструменты отладки -> Автоматически закрывать консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

#### 5) Составил блок-схему для программы.

