

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»
Вариант __

Выполнил:
Иващенко Олег Андреевич
2 курс, группа ИВТ-б-о-22-1,
09.03.02 «Информационные и
вычислительные машины»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики:
Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Порядок выполнения работы:

Задача 16:

Задан алгоритм вычисления функции $F(n)$, где n — натуральное число:

$F(n) = 1$, при $n = 1$;

$F(n) = n - 2 + F(n - 1)$, если $n > 1$.

Чему равно значение функции $F(2023) - F(2021)$?

```
using System;

class Ex_1_13
{
    static void Main()
    {
        Console.WriteLine("Введите первое натуральное число: "); int number1 =
Convert.ToInt32(Console.ReadLine());
        if (number1 < 1) Console.WriteLine("Вы ввели ненатуральное число");
        Console.WriteLine("Введите второе натуральное число: "); int number2 =
Convert.ToInt32(Console.ReadLine());
        if (number2 < 1) Console.WriteLine("Вы ввели ненатуральное число");
        int result = Rec(number1) - Rec(number2);
        Console.WriteLine("Разница: " + result.ToString());
        Console.ReadKey();
    }

    static int Rec(int Number)
    {
        if (Number == 1) return 1;
        else if (Number > 1) return Number - 2 + Rec(Number - 1);
        return -1;
    }
}
```

Таблица 1 – Код программы

```
C:\Users\PackardBell\source\repos\ConsoleApp3\ConsoleApp3\bin\Debug\ConsoleApp3.exe
Введите первое натуральное число:
2023
Введите второе натуральное число:
2021
Разница: 4041
-
```

Рисунок 1.1 – Результат выполнения программы

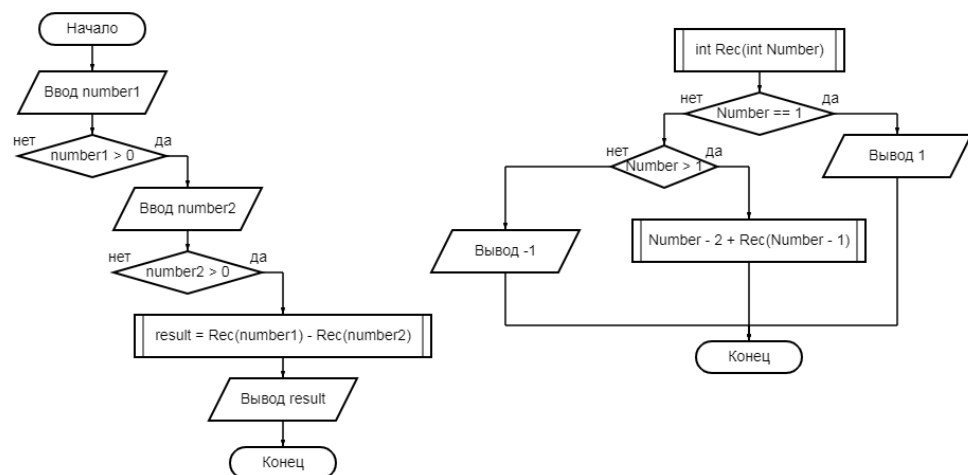


Рисунок 1.2 – Блок-схема программы

Задача 17:

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 разность элементов кратна 60, затем максимальную из разностей элементов таких пар. В данной задаче под парой подразумевается два различных элемента последовательности. Порядок элементов в паре не важен.

```
using System;

class Ex_2_17
{
    static void Main()
    {
        int[] intArray = new int[10000];
```

```

int counter = 0, max = int.MinValue, maxi = -1, maxj = -1;
Random rnd = new Random();
for (int i = 0; i < intArray.Length; i++) intArray[i] = rnd.Next(10000);

for (int i = 0; i < intArray.Length; i++)
    for (int j = 0; j < intArray.Length; j++)
        if (intArray[i] != intArray[j])
        {
            int tempInt = intArray[i] - intArray[j];
            if (tempInt % 60 == 0)
            {
                counter++;
                if (tempInt > max) { max = tempInt; maxi = i; maxj = j; }
            }
        }

Console.WriteLine("Количество пар с разностью кратной 60: " + counter.ToString());
Console.WriteLine("Максимальная разница у чисел " + intArray[maxi] + " и " + intArray[maxj] + ": " +
max);
Console.ReadKey();
}
}

```

Таблица 2 – Код программы

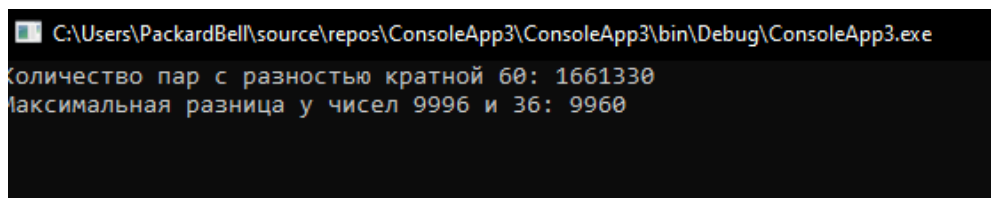


Рисунок 2.1 – Результат выполнения программы

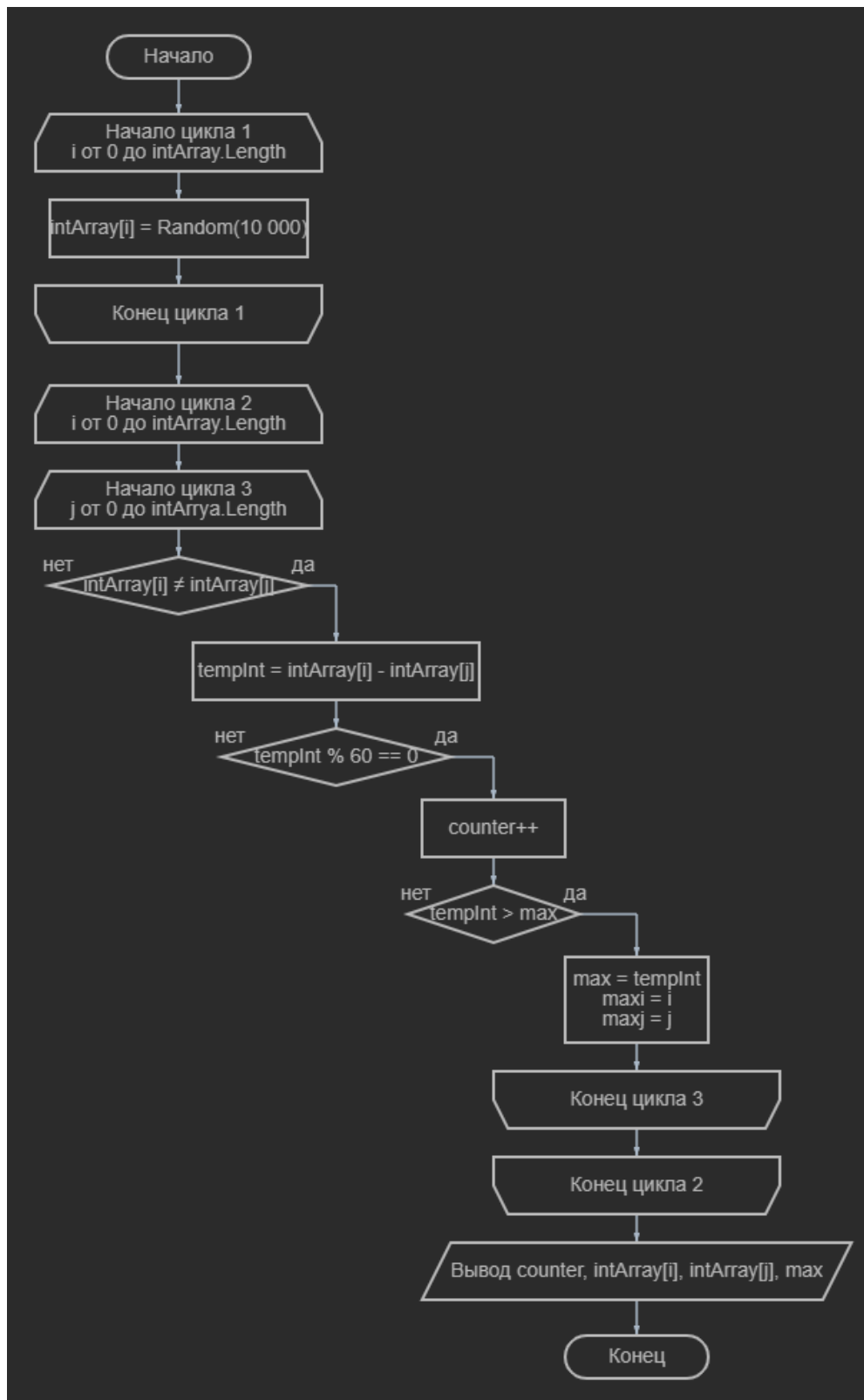


Рисунок 2.2 – Блок-схема программы

Задача:

Квадрат разлинован на $N \times N$ клеток ($1 < N < 30$). 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. Квадрат ограничен внешними стенами. Между соседними клетками квадрата также могут быть внутренние стены. Сквозь стену Робот пройти не может. 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 клеткам маршрута Робота.

Определите максимальную и минимальную денежные суммы, которые может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответе укажите два числа — сначала максимальную сумму, затем минимальную.

```
using System;
using System.IO;

namespace ConsoleApp2
{
    class Program
    {
        static void Main()
        {
            Cell[,] Field = new Cell[20, 20];
            int[,] MaxMoney = new int[20, 20];
            int[,] MinMoney = new int[20, 20];

            StreamReader Path = new StreamReader($"C:\\Users\\PackardBell\\Desktop\\L18.txt");
            string Text = Path.ReadToEnd();
            string[] Line = Text.Split('\n');
            for (int i = 0; i < 20; i++)
            {
                string[] strCell = Line[i].Split(' ');
                for (int j = 0; j < 20; j++) Field[i, j] = new Cell(int.Parse(strCell[j]));
            }

            for (int y = 1; y <= 8; y++) Field[1, y].Blocked = Cell.Block.Right;
```

```

for (int y = 4; y <= 13; y++) Field[3, y].Blocked = Cell.Block.Right;
for (int y = 9; y <= 13; y++) Field[5, y].Blocked = Cell.Block.Right;
for (int y = 5; y <= 13; y++) Field[11, y].Blocked = Cell.Block.Right;
for (int y = 8; y <= 17; y++) Field[14, y].Blocked = Cell.Block.Right;
for (int x = 3; x <= 7; x++) Field[x, 1].Blocked = Cell.Block.Bottom;
for (int x = 8; x <= 13; x++) Field[x, 16].Blocked = Cell.Block.Bottom;

```

```

for (int i = 1; i < 20; i++)

```

```

{
    MaxMoney[i, 0] = MaxMoney[i - 1, 0] + Field[i, 0].Money;
    MinMoney[i, 0] = MinMoney[i - 1, 0] + Field[i, 0].Money;

```

```

    MaxMoney[0, i] = MaxMoney[0, i - 1] + Field[0, i].Money;
    MinMoney[0, i] = MinMoney[0, i - 1] + Field[0, i].Money;

```

```

}

```

```

for (int i = 1; i < 20; i++)

```

```

{
    for (int j = 1; j < 20; j++)
    {
        if (Field[i, j].Blocked == Cell.Block.None)
        {
            MaxMoney[i, j] = Math.Max(MaxMoney[i - 1, j], MaxMoney[i, j - 1]) + Field[i, j].Money;
            MinMoney[i, j] = Math.Min(MinMoney[i - 1, j], MinMoney[i, j - 1]) + Field[i, j].Money;
        }
        else if (Field[i, j].Blocked == Cell.Block.Right)
        {
            MaxMoney[i, j] = MaxMoney[i, j - 1] + Field[i, j].Money;
            MinMoney[i, j] = MinMoney[i, j - 1] + Field[i, j].Money;
        }
        else if (Field[i, j].Blocked == Cell.Block.Bottom)
        {
            MaxMoney[i, j] = MaxMoney[i - 1, j] + Field[i, j].Money;
            MinMoney[i, j] = MinMoney[i - 1, j] + Field[i, j].Money;
        }
    }
}

```

```

Console.WriteLine("Максимальное количество монет: " + MaxMoney[14, 14].ToString());

```

```

Console.WriteLine("Минимальное количество монет: " + MinMoney[14, 14].ToString());

```

```

Console.ReadKey();

```

```

}

```

```

}

```

```

class Cell
{
    public enum Block
    {
        None,
        Right,
        Bottom
    }

    public Block Blocked;
    public int Money;

    public Cell(int Money)
    {
        Blocked = Block.None;
        this.Money = Money;
    }
}

```

Таблица 3 – Код программы

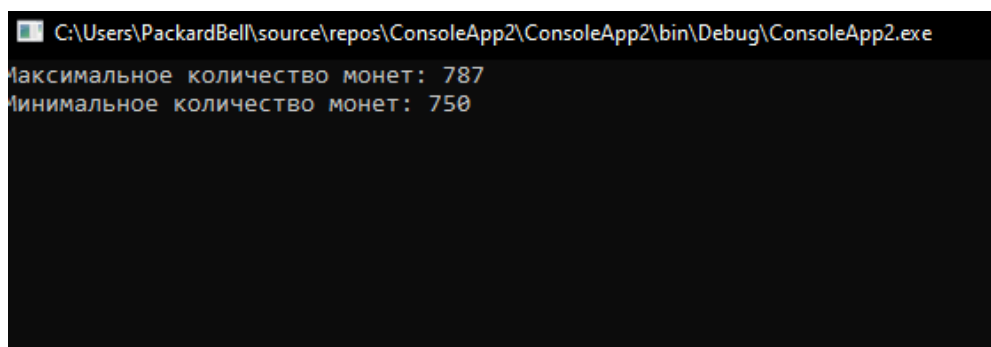


Рисунок 3.1 – Результат выполнения программы

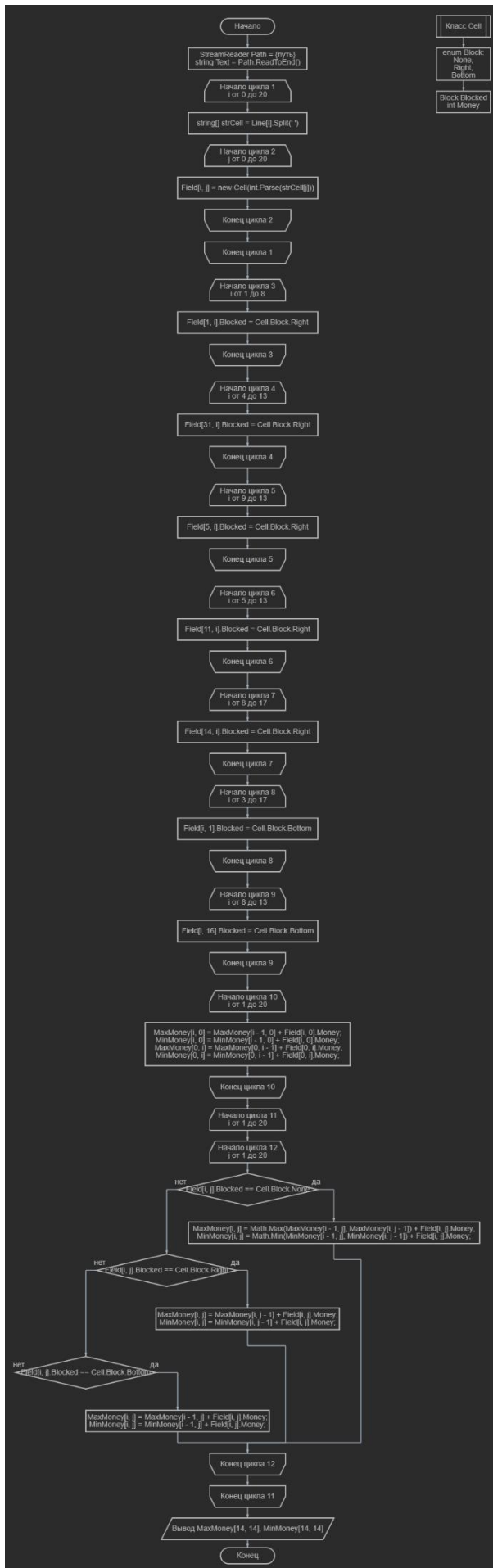


Рисунок 3.2 – Блок-схема программы