**Лабораторная работа №15**

**Рекурсивные алгоритмы**

**Цели:**

1) Познакомиться с рекурсивными алгоритмами.

2) Получить навыки разработки функций C#, реализующих вычисления по рекуррентным формулам.

**Приложение Lab11\_01. Вычисление факториала по рекуррентной формуле.**

***Задание:***

Разработать консольное приложение, содержащее функцию Fact\_Rec(), вычисляющую факториал целого неотрицательного числа n по рекуррентной формуле.

***Константы:***

Отсутствуют

***Переменные:***

n

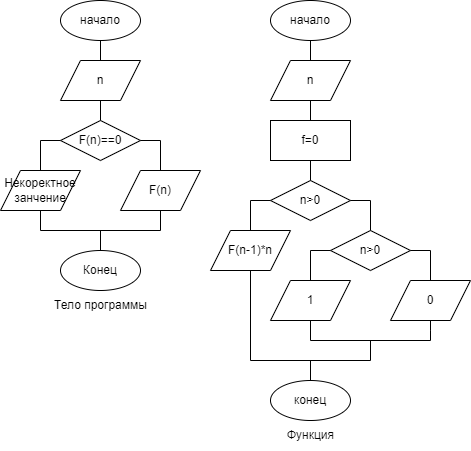
***Исходные данные:***

- n – число, факториал которого нам требуется вычислить

***Результат:***

В окне консоли выводится факториал числа.

**Блок-схема программы:**

****

***Код приложения:***

**Листинг 1 – Код консольного приложения Lab15\_01**

namespace GaivoronskayEA.OP.Lab15\_01

{

class Program

{

static int Fact\_Rec(int n)

{

if (n > 0)

{

return Fact\_Rec(n - 1) \* n;

}

else

{

if (n == 0)

{

return 1;

}

else

{

return 0;

}

}

}

static void Main(string[] args)

{

Console.WriteLine("Лабораторная работа №15");

Console.WriteLine("Приложение Lab15\_01");

Console.WriteLine("Разработчик: Гайворонская Е. А");

Console.WriteLine("-------------------------------");

Console.WriteLine("Вычисление факториала по рекуррентной формуле");

Console.WriteLine("Введите число:");

int n = int.Parse(Console.ReadLine());

if (Fact\_Rec(n) == 0)

{

Console.WriteLine("Некоректное значение:");

}

else

{

Console.WriteLine(Fact\_Rec(n));

}

}

}

}

**Тестирование:**

Пример 1)

**Исходные данные:**

n = -1

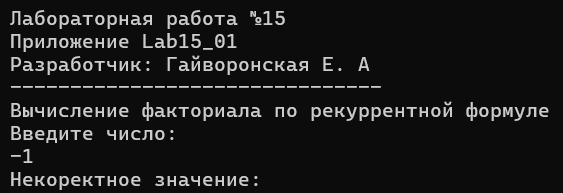


Рисунок 1 – Консольное приложение Lab15\_01

Пример 2)

**Исходные данные:**

n = 0

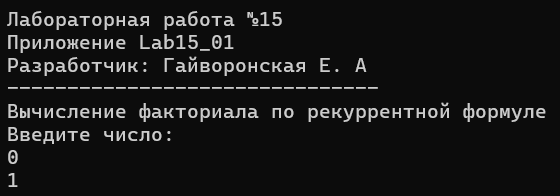


Рисунок 2 – Консольное приложение Lab15\_01

Пример 3)

**Исходные данные:**

n = 4

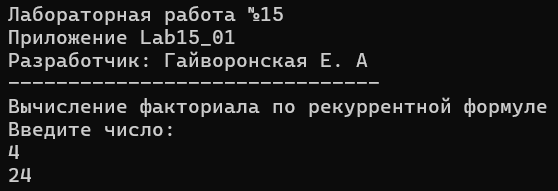


Рисунок 3 – Консольное приложение Lab15\_01

**Приложение Lab15\_02. Подсчет числа сочетаний без повторений по рекуррентной формуле.**

***Задание:***

Разработать консольное приложение, которое подсчитывало бы число сочетаний без повторений из n элементов по m мест ( n ≥ m > 0 ) по рекуррентной формуле

***Константы:***

Отсутствуют

***Переменные:***

n, m

***Исходные данные:***

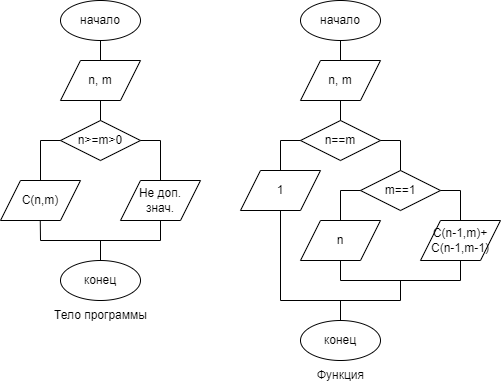
- n – число

- m – число

***Результат:***

В окне консоли выводится число сочетаний.

**Блок-схема программы:**



***Код приложения:***

**Листинг 2 – Код консольного приложения Lab15\_02**

namespace GaivoronskayEA.OP.Lab15\_02

{

class Program

{

static int C\_Rec(int n, int m)

{

if (n == m)

{

return 1;

}

else

{

if (m == 1)

{

return n;

}

else

{

{

return C\_Rec(n - 1, m) + C\_Rec(n - 1, m - 1);

}

}

}

}

static void Main(string[] args)

{

Console.WriteLine("Лабораторная работа №15");

Console.WriteLine("Приложение Lab15\_02");

Console.WriteLine("Разработчик: Гайворонская Е. А");

Console.WriteLine("-------------------------------");

Console.WriteLine("Подсчет числа сочетаний без повторений по рекуррентной формуле");

Console.WriteLine("Введите число n:");

int n = int.Parse(Console.ReadLine());

Console.WriteLine("Введите число m:");

int m = int.Parse(Console.ReadLine());

if (n >= m && m > 0)

{

Console.WriteLine(C\_Rec(n, m));

}

else

{

Console.WriteLine("Недопустимое значение");

}

}

}

}

**Тестирование:**

Пример 1)

**Исходные данные:**

n = 5

m = 5

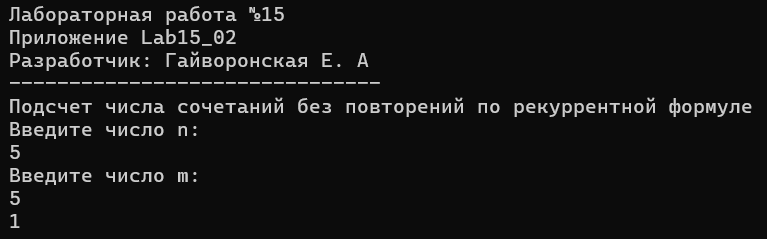


Рисунок 4 – Консольное приложение Lab15\_02

Пример 2)

**Исходные данные:**

n = 5

m = 1

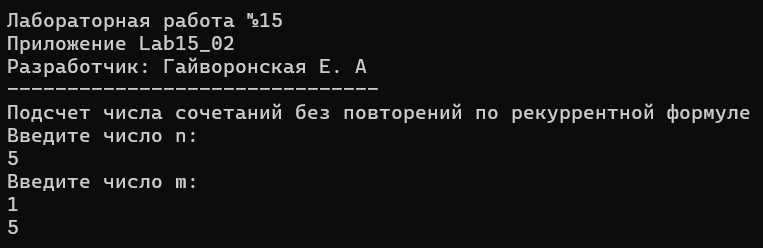


Рисунок 5 – Консольное приложение Lab15\_02

Пример 3)

**Исходные данные:**

n = 5

m = 4

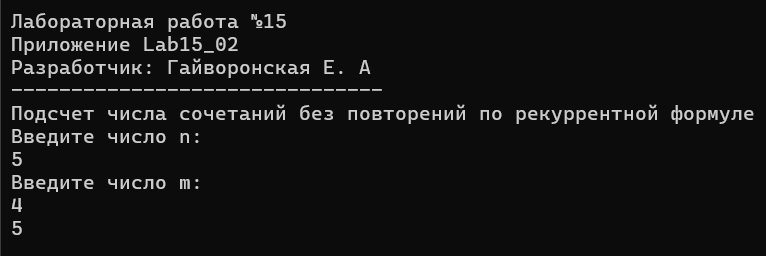


Рисунок 6 – Консольное приложение Lab15\_02

Пример 4)

**Исходные данные:**

n = -3

m = 1

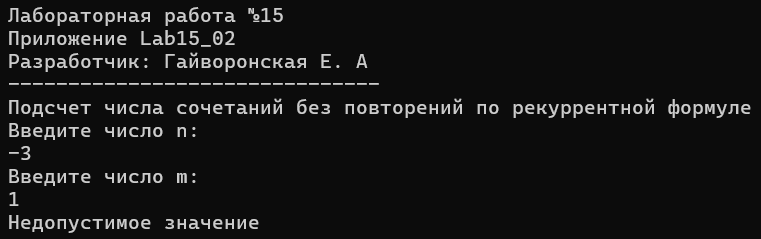


Рисунок 7 – Консольное приложение Lab15\_02