|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **БИОМЕДИЦИНСКАЯ ТЕХНИКА**

КАФЕДРА **БИОМЕДИЦИНСКИЕ ТЕХНИЧЕСКИЕ СИСТЕМЫ (БМТ-1)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика (Цифровые биомедицинские системы)**

**Отчет**

|  |
| --- |
| **по лабораторной работе № 3** |

**Название:**

Функциональная и модульная декомпозиция

**Дисциплина:** Алгоритмизация и программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | БМТ1-13Б |  |  | И.А. Атнагулов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Т.А.Ким |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

# Задание

Дано четное число N > 2. Проверить для него гипотезу Гольдбаха, состоящую в том, что каждое четное число представимо в виде суммы двух простых чисел. Вывести на экран само число и простые числа, из которых оно формируется.

# Исходный код

#Lab. №3 Main

#Вариант 2

module Main

include("Lab. №3 Include.jl")

using.Include

function main()

    println("Введите чётное целое число больше 2, я покажу из суммы каких простых чисел оно состоит.")

    n = input()

    println(primes(n))

end

using Test

@testset "mod\_test" begin

    @test primes(4) == (2,2)

    @test primes(6) == (3,3)

    @test primes(20) == (3,17)

    @test primes(22) == (3,19)

end

end

Main.main()

#Lab. №3 Include

#Вариант 2

module Include

export input,primes

    function input()

        n = parse(Int32, readline())

        return n

    end

    function eratostene(n)

        res = zeros(Int8,3\*n)

        for i in 2:n

            if res[i]==1

                continue

            end

            j = i\*2

            while j<=n

                res[j] = 1

                j+=i

            end

        end

        return res

    end

    function primes(n)

        sieve = eratostene(n)

        ans = 0

        for i in 2:n-1

            if (sieve[i] == 0)  & (sieve[n-i] == 0) & (i<=n-i)

                return i,n-i

            end

        end

        println("Введено некорректное число.")

    end

end

# Схема алгоритма



# Тестирование алгоритма

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование проверки** | **Ввод** | **Полученный результат** | **Вывод** |
| 1 | 20 | (3, 17) | Программа работает. |
| 2 | 22 | (3, 19) | Программа работает |

# Выводы

Я научился работать с модулями и юнит тестами в языке Julia