Тест Ферма

function fermat_test(n::Int, k::Int = 5)::String

```
if n <= 1
    return "Число должно быть больше 1"
end
if n == 2 || n == 3
    return "Число $n, вероятно, простое"
end
if iseven(n)
    return "Число $n составное"
end

for _ in 1:k
    a = rand(2:(n-2))  # Выбираем случайное a
    if gcd(a, n) != 1
        return "Число $n составное"
    end
    if powermod(a, n-1, n) != 1
        return "Число $n составное"
    end
end
return "Число $n, вероятно, простое"
```

end

Символ Якоби

function jacobi_symbol(a::Int, n::Int)::Int

```
if n <= 0 || iseven(n)
    error("Модуль должен быть нечетным положительным числом")
end
a = mod(a, n)
result = 1
while a != 0
    while iseven(a)
        a ÷= 2
        if mod(n, 8) ∈ [3, 5]
            result *= -1
        end
    end
    a, n = n, a
    if mod(a, 4) == 3 && mod(n, 4) == 3
```

```
        result *= -1
    end
    a = mod(a, n)
end
return n == 1 ? result : 0
```

end

Тест Соловея-Штрассена

function solovay_strassen_test(n::Int, k::Int = 5)::String

```
if n <= 1
    return "Число должно быть больше 1"
end
if n == 2 || n == 3
    return "Число $n, вероятно, простое"
end
if iseven(n)
    return "Число $n составное"
end

for _ in 1:k
    a = rand(2:(n-2))
    if gcd(a, n) != 1
        return "Число $n составное"
    end
    j = jacobi_symbol(a, n)
    if powermod(a, (n-1)÷2, n) != mod(j, n)
        return "Число $n составное"
    end
end
return "Число $n, вероятно, простое"
```

end

Тест Миллера-Рабина

function miller_rabin_test(n::Int, k::Int = 5)::String

```
if n <= 1
    return "Число должно быть больше 1"
end
if n == 2 || n == 3
    return "Число $n, вероятно, простое"
```

```
    end
    if iseven(n)
        return "Число $n составное"
    end

    # Разложение n-1 = 2^s * r
    s = 0
    r = n - 1
    while iseven(r)
        r ÷= 2
        s += 1
    end

    for _ in 1:k
        a = rand(2:(n-2))
        x = powermod(a, r, n)
        if x == 1 || x == n-1
            continue
        end
        for __ in 1:(s-1)
            x = powermod(x, 2, n)
            if x == n-1
                break
            end
        end
        if x != n-1
            return "Число $n составное"
        end
    end
    return "Число $n, вероятно, простое"
```

end

Пример использования

println("Тест Ферма:") println(fermat_test(17)) # Пример: 17 — простое println(fermat_test(341)) # Пример: 341 — составное

println("\nТест Соловея-Штрассена:") println(solovay_strassen_test(17)) # Пример: 17 — простое println(solovay_strassen_test(341)) # Пример: 341 — составное

println("\nТест Миллера-Рабина:") println(miller_rabin_test(17)) # Пример: 17 — простое println(miller_rabin_test(341)) # Пример: 341 — составное