

using Random

## Функция для вычисления наибольшего общего делителя (НОД)

---

```
function gcd(a::Int, b::Int)::Int while b != 0 a, b = b, a % b end return a end
```

## Функция $f(x) = x^2 + c \pmod n$

---

```
function f(x::Int, c::Int, n::Int)::Int return (x^2 + c) % n end
```

## Алгоритм Полларда для нахождения нетривиального делителя числа n

---

```
function pollard_rho(n::Int, c::Int = 1)::Int # Начальные значения a = c b = c d = 1
```

```
# Цикл поиска нетривиального делителя
while d == 1
    # Обновляем значения a и b
    a = f(a, c, n)          # Одинарный шаг
    b = f(f(b, c, n), c, n) # Двойной шаг
    d = gcd(abs(a - b), n)  # Вычисляем НОД

    # Если найден нетривиальный делитель
    if 1 < d < n
        return d
    elseif d == n
        error("Делитель не найден. Попробуйте другое начальное значение c.")
    end
end
```

```
end
```

## Функция для разложения числа n на множители

---

```
function factorize(n::Int)::Vector{Int} factors = Int[]
```

```
# Пока n не станет простым числом
while true
    # Проверяем, является ли n простым
    is_prime = true
    for i in 2:isqrt(n)
```

```
        if n % i == 0
            is_prime = false
            break
        end
    end

    if is_prime
        push!(factors, n)
        break
    end

    # Ищем нетривиальный делитель с помощью алгоритма Полларда
    divisor = pollard_rho(n)
    push!(factors, divisor)
    n = div(n, divisor)
end

return sort(factors)
```

end

## Пример использования

---

n = 1359331 println("Разложение числа \$n на множители: ", factorize(n))