# Отчёт по лабораторной работе №6. Разложение чисел на множители

*Дисциплина: Математические основы защиты информации и информационной безопасности*

**Студент:** Леонова Алина Дмитриевна, 1032212306
**Группа:** НФИмд-01-21
**Преподаватель:** д-р.ф.-м.н., проф. Кулябов Дмитрий Сергеевич

18 декабря, 2021, Москва

# Цель и задание работы

## Цель работы

Целью данной работы является ознакомление с методом разложения чисел на множители и реализация этого метода на выбранном языке программирования.

## Задание

1. Реализовать рассмотренный алгоритм программно.
2. Разложить на множители данное преподавателем число.

# Теоретическое введение

# $\rho$-метод Полларда

$\rho$-алгоритм предложен Джоном Поллардом в 1975 году для факторизации (разложения на множители) целых чисел. Данный алгоритм основывается на алгоритме Флойда поиска длины цикла в последовательности и некоторых следствиях из парадокса дней рождения. Алгоритм наиболее эффективен при факторизации составных чисел с достаточно малыми множителями в разложении.

Сложность алгоритма оценивается как $O(N^{1/4})$.

# $\rho$-метод Полларда

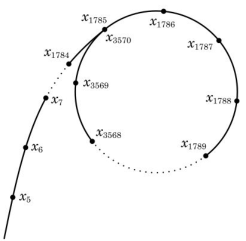



**Figure 1:** Цикл может быть представлен в виде греческой буквы ро

$\rho$-алгоритм Полларда строит числовую последовательность, элементы которой образуют цикл, начиная с некоторого номера n, что может быть проиллюстрировано, расположением чисел в виде греческой буквы $\rho$ (см. рис. 1), что послужило названием семейству алгоритмов.

# Выполнение лабораторной работы

# Промежуточные функции

```python
# Алгоритм Евклида
def nod(a, b):
    if a == 0 or b == 0:
        return max(a, b)
    if a == 1 or b == 1:
        return 1
    if a < b:
        a, b = b, a
    d = nod(a % b, b)
    return d


# Функция
def eval_(f, x, n):
    return eval(f)
```

**Figure 2:** Промежуточные функции

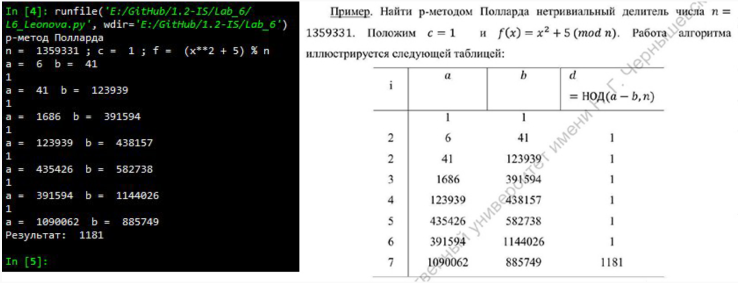
Функция для нахождения наибольшего общего делителя a и b - Алгоритм Евклида, и функция eval_ для нахождения результата переданной как строки функции f с переданными аргументами x и n (см. рис. 2).

# Реализация $\rho$-метода Полларда

```python
# р-метод Полларда
def Pollard(n, c, f):
    print('n = ', n, '; c = ', c,'; f = ', f)
    a, b = c, c

    while True:
        a = eval_(f, a, n) % n
        b = eval_(f, eval_(f, b, n), n) % n
        print('a = ',a,' b = ',b)

        if a - b < 0:
            d = 1
        else:
            d = nod(a-b, n)

        if 1 < d and d < n:
            return d
        if d == n:
            return print('Делитель не найден')
        if d == 1:
            print('1')


print('р-метод Полларда')
print('Результат: ',Pollard(1359331, 1, '(x**2 + 5) % n'))
```

**Figure 3:** Функция ро-метода Полларда

# Результаты



```
In [4]: runfile('E:/GitHub/1.2-IS/Lab_6/L6_Leonova.py', wdir='E:/GitHub/1.2-IS/Lab_6')
p-метод Полларда
n =  1359331 ; c =  1 ; f =  (x**2 + 5) % n
a =  6  b =  41
1
a =  41  b =  123939
1
a =  1686  b =  391594
1
a =  123939  b =  438157
1
a =  435426  b =  582738
1
a =  391594  b =  1144026
1
a =  1090062  b =  885749
Результат:  1181

In [5]:
```

Пример. Найти p-методом Полларда нетривиальный делитель числа $n = 1359331$. Положим $c = 1$ и $f(x) = x^2 + 5 \ (mod\ n)$. Работа алгоритма иллюстрируется следующей таблицей:

| i | $a$ | $b$ | $d = \text{НОД}(a-b, n)$ |
|---|---|---|---|
| | 1 | 1 | |
| 2 | 6 | 41 | 1 |
| 2 | 41 | 123939 | 1 |
| 3 | 1686 | 391594 | 1 |
| 4 | 123939 | 438157 | 1 |
| 5 | 435426 | 582738 | 1 |
| 6 | 391594 | 1144026 | 1 |
| 7 | 1090062 | 885749 | 1181 |

**Figure 4:** Результат выполнения L6_Leonova.py и задание

Результат выполнения программы, проверка реализации $\rho$-метода Полларда, разложение на множители данного в задании числа (см. рис. 4).

# Выводы

Цель лабораторной работы была достигнута, метод разложения чисел на множители - $\rho$-Метод Полларда - был реализован на языке программирования Python.