# Отчёт по лабораторной работе №5. Вероятностные алгоритмы проверки чисел на простоту

*Дисциплина: Математические основы защиты информации и информационной безопасности*

**Студент:** Агеева Анастасия Сергеевна, 1032212304
**Группа:** НФИмд-02-21
**Преподаватель:** д-р.ф.-м.н., проф. Кулябов Дмитрий Сергеевич

11 декабря, 2021, Москва

# Прагматика

# Прагматика данной лабораторной работы

- В рамках дисциплины “Математические основы защиты информации и информационной безопасности” нам необходимо изучить ее разделы.
- Данная работа необходима для более глубоко и детального понимания работы алгоритмов шифрования.

# Цель

# Цель выполнения данной лабораторной работы

- Цель данной лабораторной работы изучение алгоритмов проверки чисел на простоту.

# Задачи

# Задачи выполнения данной лабораторной работы

1. Реализовать программно алгоритм, реализующий тест Ферма;
2. Реализовать алгоритм вычисления символа Якоби;
3. Реализовать программно алгоритм, реализующий тест Соловэя-Штрассена;
4. Реализовать программно алгоритм, реализующий тест Миллера-Рабина.

# Результаты выполнения данной лабораторной работы

# Тест Ферма

```
In [2]:
def ferma(n):
    if n < 5:
        return "Введите корректные  данные"
    a = random.randint(2, n-2)
    r = a**(n-1)%n
    if r == 1:
        return "Число n, вероятно, простое"
    else: return "Число n составное"
```

**Figure 1:** Реализация теста Ферма

```
In [3]:
t = ferma(8)
t

Out[3]: 'Число n составное'

In [4]:
t = ferma(7)
t

Out[4]: 'Число n, вероятно, простое'
```

**Figure 2:** Результаты теста Ферма

# Алгоритм вычисления символа Якоби

In [5]:
```
def yakobi(a, n):
    if n < 3 or a >= n or a < 0:
        print ("Введите корректные  данные")
        return
    g = 1
    while True:
        if a == 0:
            return 0
        if a == 1:
            return g
        k = 0
        a1 = a
        while a1 % 2 == 0:
            k += 1
            a1 /= 2
        if k % 2 == 0:
            s = 1
        elif (n-1)%8 == 0 or (n+1)%8 == 0:
            s = 1
        elif (n-3)%8 == 0 or (n+3)%8 == 0:
            s = -1
        if a1 == 1:
            return s*g
        else:
            if (n - 3)%4 == 0 and (a1 - 3)%4 == 0:
                s = -s
        a = n%a1
        n = a1
        g = g*s
```

**Figure 3:** Реализация алгоритма вычисления символа Якоби

In [6]:
```
y = yakobi(4, 13)
print("Символ Якоби:", y)
```

```
Символ Якоби: 1
```

In [7]:
```
y = yakobi(4, 2)
print("Символ Якоби:", y)
```

```
Введите корректные  данные
Символ Якоби: None
```

**Figure 4:** Результаты алгоритма вычисления символа Якоби

# Тест Соловэя-Штрассена

```
In [8]:
def solovey(n):
    if n < 5 or n%2 == 0:
        return "Введите корректные  данные"
    a = random.randint(2, n-3) # включительно границы
    r = (a**int((n-1)/2))%n
    if r != 1 and r != n-1:
        return "Число n составное"
    s = yakobi(a, n)
    if (r-s)%n != 0:
        return "Число n составное"
    else:
        return "Число n, вероятно, простое"
```

**Figure 5:** Реализация теста Соловэя-Штрассена

```
In [9]:
t = solovey(11)
t

Out[9]: 'Число n, вероятно, простое'

In [10]:
t = solovey(9)
t

Out[10]: 'Число n составное'
```

**Figure 6:** Результаты теста Соловэя-Штрассена

# Тест Миллера-Рабина

In [11]:
```
def miller(n):
    if n < 5 or n%2 == 0:
        return "Введите корректные  данные"
    tmp = n-1
    s = 0
    r = tmp
    while r % 2 == 0:
        s += 1
        r /= 2
    a = random.randint(2, n-3)
    y = (a**r)%n
    if y != 1 and y != n-1:
        j = 1
        while j <= s-1 and y != n-1:
            y = y**2%n
            if y == 1:
                return "Число n составное"
            j+=1
        if y != n-1:
            return "Число n составное"
    return "Число n, вероятно, простое"
```

**Figure 7:** Реализация теста Миллера-Рабина

In [12]:
```
t = miller(9)
t
```

Out[12]: 'Число n составное'

In [13]:
```
t = miller(11)
t
```

Out[13]: 'Число n, вероятно, простое'

**Figure 8:** Результаты теста Миллера-Рабина

# Выводы

- Исходя из теоретических сведений, программы выполнены без ошибок, чему свидетельствуют полученные результаты.
- В ходе данной лабораторной работы я реализовала три алгоритма проверки числа на простоту, а также алгоритм нахождения числа Якоби.