# Прагматика выполнения лабораторной работы

Студенты должны разбираться в методах шифрования и познакомиться способом шифрования текста гаммированию. Поэтому освоение и реализацию нахождения НОД разнообразными методами Евклида положительно скажется на будущее понимание процесса шифрования.

# Цель выполнения лабораторной работы

Реализовать с помощью программирования программы нахождения НОД, методами, описанными в задании к лабораторной работе №4.

# Задачи выполнения лабораторной работы

Разработать 4 программы, которые будут представлять из себя:

1. Алгоритм Евклида.
2. Расширенный алгоритм Евклида.
3. Бинарный алгоритм Евклида.
4. Расширенный Бинарный алгоритм Евклида.

# Результаты выполнения лабораторной работы

1. Реализовала программу поиска НОД методом алгоритма Евклида.

(рис. -@fig:001)

```
d=int(input("Введите 1 число"))
k=int(input("Введите 2 число"))
#d, a - большее число
#k, b - меньшее

if k>d:
  a,b = d,k
else:
  a,b = k,d

c=100

while c!=0:
  c=a%b
  if c!=0:
    a,b=b,c

print("НОД (", d,", ",k,") = ",b)
```

```
Введите 1 число15
Введите 2 число3
НОД ( 15 ,  3 ) =  3
```

Основная суть алгоритма лежит в том, чтобы большее число делить на меньшее и сохранять остатки от деления для дальнейших операций. А самим НОД будет делитель в том случае, когда остатка от деления не останется.

# Результаты выполнения лабораторной работы

2. Реализовала программу расширенного алгоритма Евклида.

(рис. -@fig:003)

```
a=int(input("Введите 1 число:"))
b=int(input("Введите 2 число:"))

def gcdex(a, b):
    if b == 0:
        return a, 0, 1
    else:
        d, x, y = gcdex(b, a % b)
        return d, y, x - y * (a % b)
d,y,x=gcdex(a, b)

print("НОД (", a,", ",b,") = ",d)
print("x : ", x,", ","y : ",y)
```

```
Введите 1 число:12
Введите 2 число:4
НОД ( 12 ,  4 ) =  4
x :  0 ,  y :  1
```

Основное отличие расширенного алгоритма заключается в поиске таких коэффициентов x и y, которые удовлетворяют условию $xa+yb=d$, где d= делитель.

# Результаты выполнения лабораторной работы

3. Реализовала бинарный алгоритм Евклида. (рис. -@fig:005)

```
A=int(input("Введите 1 число: "))
B=int(input("Введите 2 число: "))
def binary_gcd(A, B):
    k = 1
    while (A != 0) and (B != 0):
        if A > B:q = a // b
        else:q = b // a

        while (A % 2 == 0) and (B % 2 == 0): A /= 2; B /= 2; k *= 2
        while A % 2 == 0: A /= 2
        while B % 2 == 0: B /= 2

        if A >= B: A -= B
        else: B -= A
    return B * k

d=binary_gcd(A, B)

print("НОД (", A,", ",B,") = ",d)
```

```
Введите 1 число:15
Введите 2 число:3
НОД ( 15 ,  3 ) =  3
```

От стандартного алгоритма его отличает использование в коде приемов со свойствами НОД. И так как бинарный алгоритм в основном использует манипуляции с четными числами, то данные вычисления проходят намного быстрее в компьютере из-за его двоичного кода.

# Результаты выполнения лабораторной работы

4. Реализовала расширенный алгоритм Евклида с бинарными вычислениями.

(рис. -@fig:003)

```python
def ext_gcd(a, b):
    if a == 0: return 1, 1, b
    if b == 0: return 1, 1, a
    if not a&1 | b&1:
        # оба чётные - x и y не трогаем, gcd удваеваем
        x, y, g = ext_gcd(a>>1, b>>1)
        return x, y, g<<1
    elif not a&1:
        # a - чётное, b - нечётное
        x, y, g = ext_gcd(a>>1, b)
        return (x-b>>1, y+(a>>1), g) if x&1 else (x>>1, y, g)
    elif not b&1:
        # a - нечётное, b - чётное
        x, y, g = ext_gcd(a, b>>1)
        return (x+(b>>1), y-a>>1, g) if y&1 else (x, y>>1, g)
    elif b > a:
        # оба нечётные
        x, y, g = ext_gcd(a, b-a)
        return x-y, y, g
    else:
        # оба нечётные
        x, y, g = ext_gcd(a-b, b)
        return x, y-x, g

a,b = 36,54
x,y,g=ext_gcd(a,b)
print(f"({x})*{a} + ({y})*{b} = {g}") # (-13)*36 + (9)*54 = 18
```

```
(-13)*36 + (9)*54 = 18
```

Являет собой квинтэссенцию всех пунктов выше, так как включает в себя и нахождение коэффициентов x и y для уравнения $xa+yb=d$, и задействованные свойства НОД с бинарными вычислениями. Убойная штука, всем советую.

# Вывод

Освоен на практике написание всех этих горемычных пунктов. Познала Дзен, ушла в буддизм.

# {.standout}Спасибо за внимание