Отчет по лабораторной работе №4

Дисциплина: Математические основы защиты информации и информационной безопасности

Выполнила Дяченко Злата Константиновна, НПМмд-02-22

Содержание

# 1 Цель работы

Ознакомится и реализовать алгоритмы вычисления наибольшего общего делителя.

# 2 Задание

Реализовать такие алгоритмы вычисления наибольшего общего делителя, как алгоритм Евклида, бинарный алгоритм Евклида, расширенный алгоритм Евклида и расширенный бинарный алгоритм Евклида.

# 3 Теоретическое введение

Пусть числа и целые и . Разделить на с остатком – значит представить в виде ,где и Число называется неполным частным, число – неполным остатком от деления на . Целое число называется наибольшим общим делителем целых чисел (обозначается d = НОД()), если выполняются следующие условия: 1. каждое из чисел делится на ;  
2. если – другой общий делитель чисел ܽ, то делится на .

Для вычисления наибольшего общего делителя двух целых чисел применяется способ повторного деления с остатком, называемый алгоритмом Евклида.

1. Алгоритм Евклида.  
   *Вход*. Целые числа .  
   *Выход*. .
   1. Положить .
   2. Найти остаток от деления на .
   3. Если , то положить . В противном случае положить и вернуться на шаг 2.
   4. Результат: .
2. Бинарный алгоритм Евклида.  
   *Вход*. Целые числа .  
   *Выход*. .
   1. Положить .
   2. Пока оба числа и четные, выполнять до получения хотя бы одного нечетного значения или .
   3. Положить .
   4. Пока выполнять следующие действия:
      1. Пока четное, полагать .
      2. Пока четное, полагать .
      3. При положить $u ← u - v $. В противном случае положить $v ← v - u $.
   5. Положить .
   6. Результат: .
3. Расширенный алгоритм Евклида.  
   *Вход*. Целые числа .  
   *Выход*. ; такие целые числа , что .
   1. Положить .
   2. Разделить с остатком на : .
   3. Если , то положить . В противном случае положить и вернуться на шаг 2.
   4. Результат: .
4. Расширенный бинарный алгоритм Евклида.  
   *Вход*. Целые числа .  
   *Выход*. .
   1. Положить .
   2. Пока оба числа и четные, выполнять до получения хотя бы одного нечетного значения или .
   3. Положить .
   4. Пока выполнять следующие действия:
      1. Пока четное:
         1. Положить .
         2. Если оба числа и четные, то положить . В противном случае положить .
      2. Пока четное:
         1. Положить .
         2. Если оба числа и четные, то положить . В противном случае положить .
      3. При положить . В противном случае положить .
   5. Положить .
   6. Результат: .

# 4 Выполнение лабораторной работы

## 4.1 Шаг 1

Ознакомилась с предоставленными теоретическими данными. Для выполнения задания решила использовать язык Python. Написала функцию, выполняющую поиск НОД с помощью алгоритма Евклида. Код функции и результат ее использования представлен на Рисунке 1 (рис. - fig. 1). Функция принимает на вход числа и . При условии, что реализуется алгоритм, представленный в теоретическом введении и функция возвращает НОД. Если условие не выполняется, функция ничего не вернет, будет выведено соответствующее сообщение. Пример выполнения функции также показан на Рисунке 1 (рис. - fig. 1).

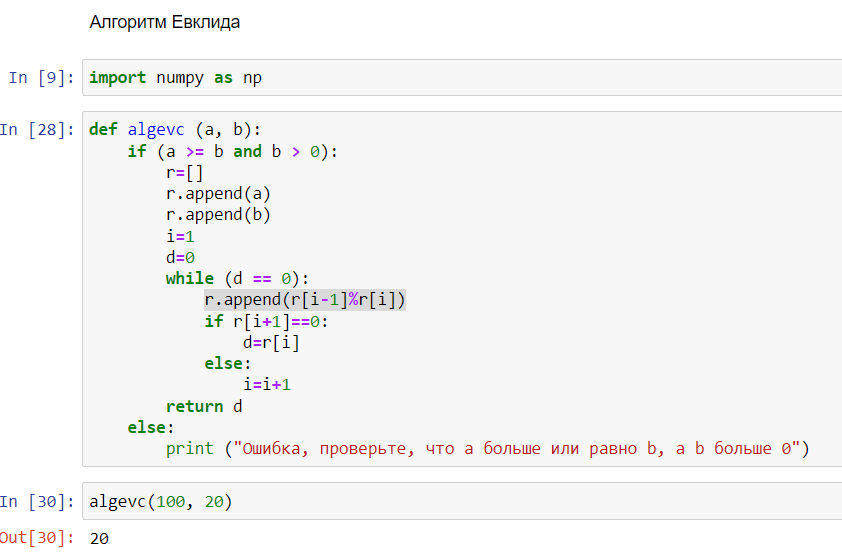


Figure 1: Реализация алгоритма Евклида

## 4.2 Шаг 2

На Рисунке 2 (рис. - fig. 2) представлен код функции, реализующий бинарный алгоритм Евклида, и пример выполнения.

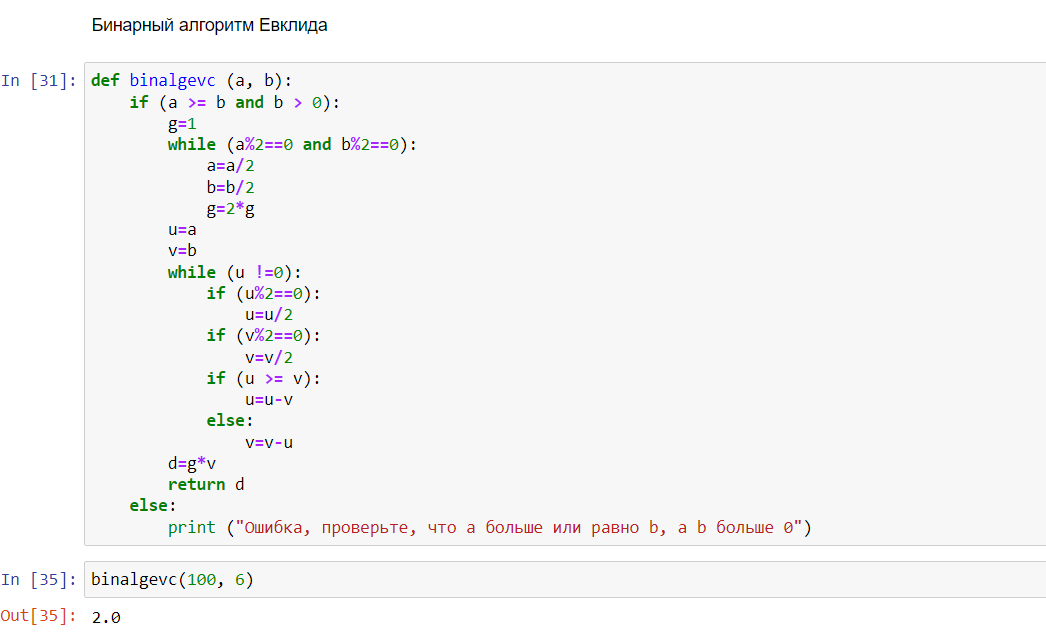


Figure 2: Реализация бинарного алгоритма Евклида

## 4.3 Шаг 3

На Рисунке 3 (рис. - fig. 3) представлен код функции, реализующий расширенный алгоритм Евклида, и пример выполнения. Данная функция в случае нахождения НОД выводит не только сам НОД, но и числа и , являющиеся коэффициентами уравнения .

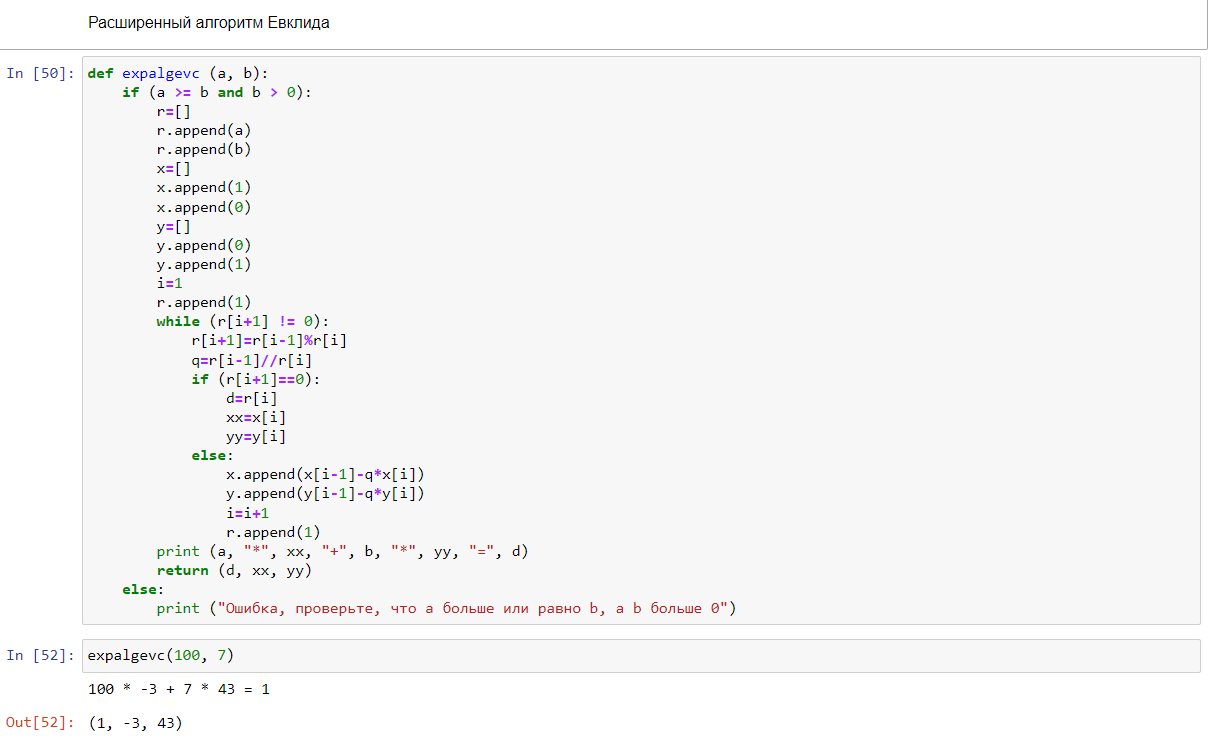


Figure 3: Реализация расширенного алгоритма Евклида

## 4.4 Шаг 4

На Рисунке 4 (рис. - fig. 4) и Рисунке 5 (рис. - fig. 5) представлен код функции, реализующий расширенный бинарный алгоритм Евклида, и пример выполнения. Данная функция в случае нахождения НОД выводит не только сам НОД, но и числа и , являющиеся коэффициентами уравнения .

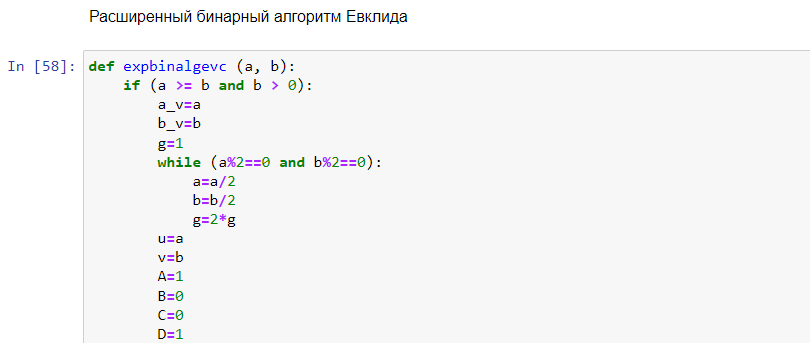


Figure 4: Реализация расширенного бинарного алгоритма Евклида

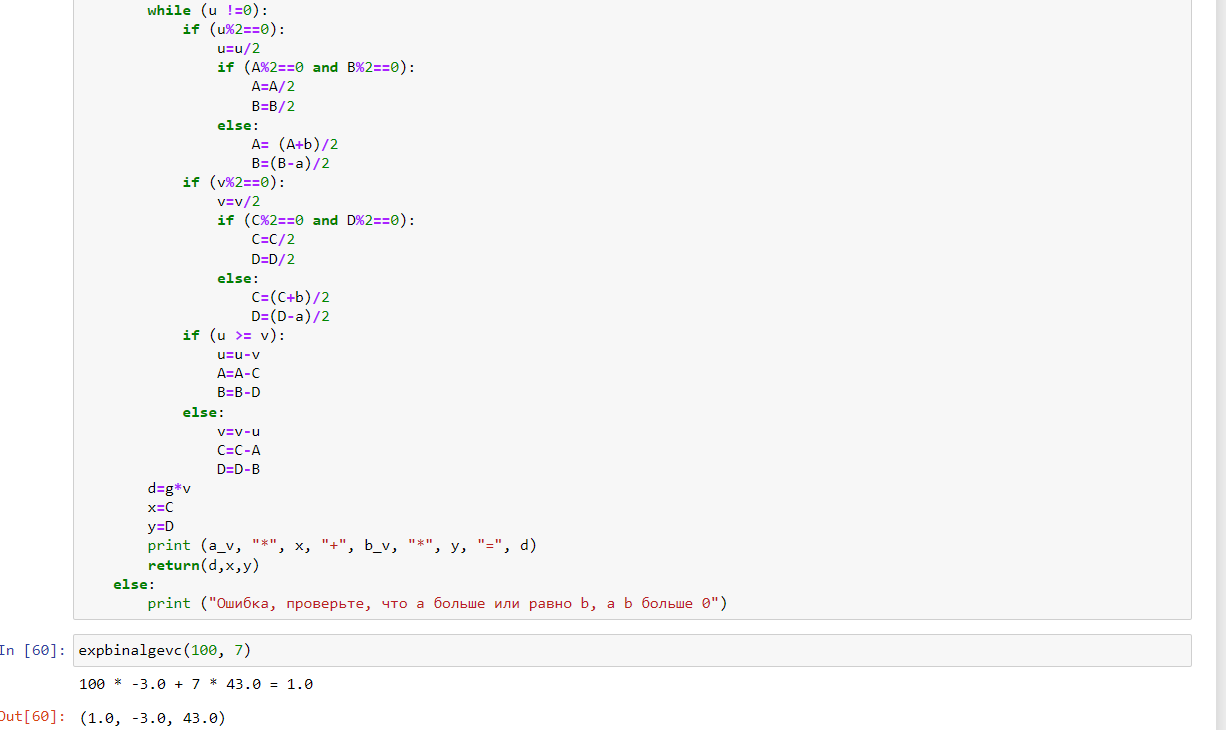


Figure 5: Реализация расширенного бинарного алгоритма Евклида

# 5 Выводы

Я ознакомилась с алгоритмами нахождения НОД и реализовала их. Результаты работы находятся в [репозитории на GitHub](https://github.com/ZlataDyachenko/workD), а также есть [скринкаст выполнения лабораторной работы](https://www.youtube.com/watch?v=H_SXHqZuYsM).