**Task 01**

**Pseudo code**

Class EuclideanAlgorithm:

Constructor:

Input: a, b (integers)

Set self.a to a

Set self.b to b

Method find\_gcd:

Start

Ensure self.a >= self.b

While self.b is not equal to 0:

Set remainder to self.a modulo self.b

Set self.a to self.b

Set self.b to remainder

Return self.a as the GCD

Main program:

Initialize num1 and num2 with integer values

Create an instance of EuclideanAlgorithm with num1 and num2

Call the find\_gcd method of the instance and store the result

Print the GCD

**Task 06**

**Pseudo Code for Extended Euclidean Algorithm**

**"""**

**PSEUDO CODE**

**function extended\_gcd(a, b):**

**if b == 0:**

**return a, 1, 0**

**else:**

**gcd, x1, y1 = extended\_gcd(b, a % b)**

**x = y1**

**y = x1 - (a // b) \* y1**

**return gcd, x, y**

**"""**

**Task 05 ( GIT )**

**It is an open source version tool to push code to the profile where organization can access or keep track of code.**

**Steps**

**1.Repository name:** [**euclidean\_algorithm**](https://github.com/kaagit1/euclidean_algorithm) **(Made on github)**

**2.Git add euclidean\_algo.py Euclidean-algo\_git ( both .py and doc file staged and pushed)**

**3.Git status (to check if the file has been staged or not)**

**4.Git commit -m “pushing euclidean algorithm:**

**5.Git push**

**Commit id: 635088a**