

# CPSC-354 Report

Eleas Vrahnos  
Chapman University

September 2, 2022

## Abstract

To be written at a later date.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Homework</b>	<b>1</b>
2.1	Week 1 . . . . .	1
<b>3</b>	<b>Project</b>	<b>3</b>
<b>4</b>	<b>Conclusions</b>	<b>3</b>

## 1 Introduction

To be written at a later date.

## 2 Homework

This section will contain my solutions to the weekly homework assignments.

### 2.1 Week 1

The following is a Python implementation of the Euclidean algorithm:

---

```
def gcd(a,b):  
    while a != b:  
        if a > b:  
            a = a-b  
        else:  
            b = b-a  
    return a
```

---

We can test this code by going through the function with a sample input `gcd(9, 33)`, step by step.

1. `gcd(9, 33)`
  - The function is called, assigning 9 to variable `a` and 33 to variable `b`.
2. `while a != b:`
  - The while loop condition returns True, so the loop starts.
3. `else:`
  - `a > b` (`9 > 33`) returns False, so the else block executes.
4. `b = b-a`
  - `b` is now assigned to  $33 - 9$ , which is 24.
5. `while a != b:`
  - The while loop condition returns True, so the loop starts.
6. `else:`
  - `a > b` (`9 > 24`) returns False, so the else block executes.
7. `b = b-a`
  - `b` is now assigned to  $24 - 9$ , which is 15.
8. `while a != b:`
  - The while loop condition returns True, so the loop starts.
9. `else:`
  - `a > b` (`9 > 15`) returns False, so the else block executes.
10. `b = b-a`
  - `b` is now assigned to  $15 - 9$ , which is 6.
11. `while a != b:`
  - The while loop condition returns True, so the loop starts.
12. `if a > b:`
  - `a > b` (`9 > 6`) returns True, so the first block executes.
13. `a = a-b`
  - `a` is now assigned to  $9 - 6$ , which is 3.
14. `while a != b:`
  - The while loop condition returns True, so the loop starts.
15. `else:`
  - `a > b` (`3 > 6`) returns False, so the else block executes.
16. `b = b-a`
  - `b` is now assigned to  $6 - 3$ , which is 3.
17. `while a != b:`
  - The while loop condition returns False (`3 == 3`), so the loop ends.
18. `return a`
  - `a` is returned from the function, giving the correct greatest common divisor of **3**.

### **3 Project**

To be written at a later date.

### **4 Conclusions**

To be written at a later date.