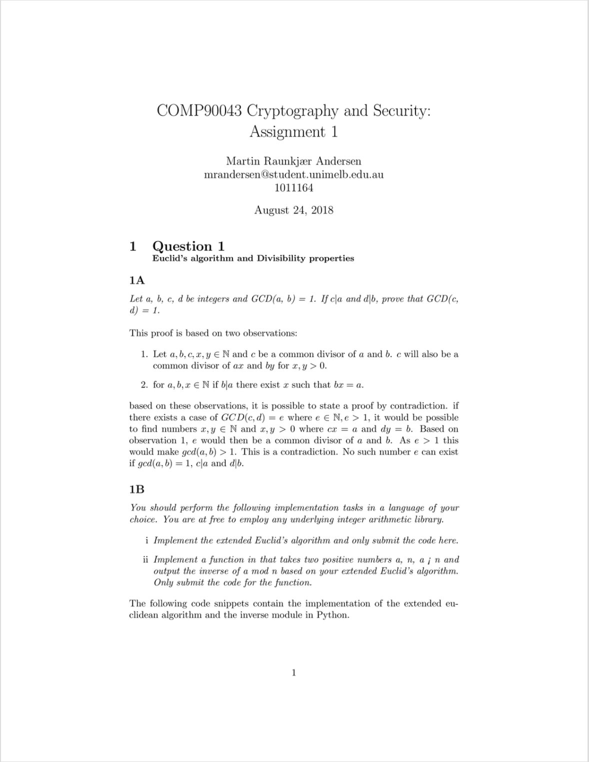
Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Martin Andersen
Assignment title:	Assignment 1 Submission
Submission title:	COMP90043_2018_SM2_Assignme..
File name:	security_assignment_1.pdf
File size:	189.81K
Page count:	6
Word count:	1,523
Character count:	6,903
Submission date:	24-Aug-2018 01:09PM (UTC+1000)
Submission ID:	992676825



COMP90043 Cryptography and Security:
Assignment 1

Martin Raunkjær Andersen
mrandersen@student.unimelb.edu.au
1011164

August 24, 2018

1 Question 1

Euclid's algorithm and Divisibility properties

1A

Let a, b, c, d be integers and GCD(a, b) = 1. If c|a and d|b, prove that GCD(c, d) = 1.

This proof is based on two observations:

1. Let $a, b, c, x, y \in \mathbb{N}$ and c be a common divisor of a and b. c will also be a common divisor of ax and by for $x, y > 0$.
2. for $a, b, x \in \mathbb{N}$ if $b|a$ there exist x such that $bx = a$.

based on these observations, it is possible to state a proof by contradiction. if there exists a case of $GCD(c, d) = e$ where $e \in \mathbb{N}, e > 1$, it would be possible to find numbers $x, y \in \mathbb{N}$ and $x, y > 0$ where $cx = a$ and $dy = b$. Based on observation 1, e would then be a common divisor of a and b. As $e > 1$ this would make $gcd(a, b) > 1$. This is a contradiction. No such number e can exist if $gcd(a, b) = 1$, $c|a$ and $d|b$.

1B

You should perform the following implementation tasks in a language of your choice. You are at free to employ any underlying integer arithmetic library.

i *Implement the extended Euclid's algorithm and only submit the code here.*

ii *Implement a function in that takes two positive numbers a, n, a ¡ n and output the inverse of a mod n based on your extended Euclid's algorithm. Only submit the code for the function.*

The following code snippets contain the implementation of the extended euclidean algorithm and the inverse module in Python.