Discrete Math

Main project

Report

Student:

John Carter

Teacher:

Bella Nicolson

Course:

Discrete Mathematics

# Program 01: Prime Factorization

**Input**: 30

**Output**: [ 2, 3, 5 ]

**Input**: 31

**Output**: [ 31 ]

**Input**: 487

**Output**: [ 487 ]

**Input**: 8893

**Output**: [ 8893 ]

**Input**: 987654323

**Output**: [ 987654323 ]

**Input**: 131317171919

**Output**: [ 19, 19, 101, 3601579 ]

# Program 02: Extended Euclidean Algorithm

## (i.)

**Input:**

a = 8359, b = 4962

**Output:**

GCD (8359 ,4962) = 1

x = -1877

y = 3162

**Input:**

a = 95,243, b = 24,138

**Output:**

GCD (95243 ,24138) = 1

x = 461

y = -1819

## (ii.)

88243*x* + 16947*y* =1

**Input:**

a = 88243, b = 16947

**Output:**

x = -2372

y = 12351

# Program 03: RSA Encryption

## (i.)

**Input:**

*P* = 44, *n* = 1,517, *e* = 49

**Output:**

The cipher text is: 1069

## (ii.)

*Choose option 2 in RSA category for using primes.*

**Input:**

**Primes:** *p* = 153,817, *q* = 1,542,689

**Exponent:** *e* = 202, 404,606

**Plain Text:** *P* = 888,999,000

**Output:**

The cipher text is: 825453345

# Program 04: RSA Decryption

## (i.)

**Input:**

*C* = 1069, *n* = 1,517 and *d* = 529

**Output:**

The plain text is: 44

## (ii.)

As we know,

ed mod ɸ(n) = 1

ɸ(n) = 237290097408

so,

using extended Euclidean Algorithm, I wrote a small piece of code, a method named private key which takes e and ɸ(n) as parameters and calculates d.

**Input:**

C = 825,453,345

d = 4,940,599,030

n = p x q = 237,319,480,973

**Output:**

Exception

## (iii.)

The decryption program does not work for Alice’s encrypted message because the public exponent generated was way too larger to be processed. There were so many possible combinations as the modulus was very large and computer got stuck in it.