```
Test Preview: Summary for hj1423 of c1
PPT 4
---------------------------------------

  Public Tests:
    student-tests/crypto-test/crypto/part 1/gcd:                 8 / 8
    student-tests/crypto-test/crypto/part 1/phi:                 9 / 9
    student-tests/crypto-test/crypto/part 1/modPow:              11 / 11
    student-tests/crypto-test/crypto/part 1/computeCoeffs:       6 / 6
    student-tests/crypto-test/crypto/part 1/inverse:             7 / 7
    student-tests/crypto-test/crypto/part 1/smallestCoPrimeOf:   6 / 6
    student-tests/crypto-test/crypto/part 1/genKeys:             6 / 6
    student-tests/crypto-test/crypto/part 1/rsaEncrypt:          4 / 4
    student-tests/crypto-test/crypto/part 1/rsaDecrypt:          4 / 4
    student-tests/crypto-test/crypto/part 2/toInt:               3 / 3
    student-tests/crypto-test/crypto/part 2/toChar:              3 / 3
    student-tests/crypto-test/crypto/part 2/add:                 3 / 3
    student-tests/crypto-test/crypto/part 2/subtract:            3 / 3
    student-tests/crypto-test/crypto/part 2/ecbEncrypt:          4 / 4
    student-tests/crypto-test/crypto/part 2/ecbDecrypt:          4 / 4
    student-tests/crypto-test/crypto/part 2/cbcEncrypt:          4 / 4
    student-tests/crypto-test/crypto/part 2/cbcDecrypt:          4 / 4
    original-tests/crypto-test/crypto/part 1/gcd:                8 / 8
    original-tests/crypto-test/crypto/part 1/phi:                9 / 9
    original-tests/crypto-test/crypto/part 1/modPow:             11 / 11
    original-tests/crypto-test/crypto/part 1/computeCoeffs:      6 / 6
    original-tests/crypto-test/crypto/part 1/inverse:            7 / 7
    original-tests/crypto-test/crypto/part 1/smallestCoPrimeOf:  6 / 6
    original-tests/crypto-test/crypto/part 1/genKeys:            6 / 6
    original-tests/crypto-test/crypto/part 1/rsaEncrypt:         4 / 4
    original-tests/crypto-test/crypto/part 1/rsaDecrypt:         4 / 4
    original-tests/crypto-test/crypto/part 2/toInt:              3 / 3
    original-tests/crypto-test/crypto/part 2/toChar:             3 / 3
    original-tests/crypto-test/crypto/part 2/add:                3 / 3
    original-tests/crypto-test/crypto/part 2/subtract:           3 / 3
    original-tests/crypto-test/crypto/part 2/ecbEncrypt:         4 / 4
    original-tests/crypto-test/crypto/part 2/ecbDecrypt:         4 / 4
    original-tests/crypto-test/crypto/part 2/cbcEncrypt:         4 / 4
    original-tests/crypto-test/crypto/part 2/cbcDecrypt:         4 / 4

Git Repo: git@gitlab.doc.ic.ac.uk:lab2324_autumn/haskellcrypto_hj1423.git
Commit ID: 535f4
```

8/10 Great Job! ☺

Please don't use magic numbers and fix the comments and all correct!.

```haskell
module Crypto ( gcd, smallestCoPrimeOf, phi, computeCoeffs, inverse
              , modPow, genKeys, rsaEncrypt, rsaDecrypt, toInt, toChar
              , add, subtract, ecbEncrypt, ecbDecrypt
              , cbcEncrypt, cbcDecrypt ) where

import Data.Char

import Prelude hiding (gcd, subtract)
import Text.Read (Lexeme(Char))

{-
The advantage of symmetric encryption schemes like AES is that they are efficient
and we can encrypt data of arbitrary size. The problem is how to share the key.
The flaw of the RSA is that it is slow and we can only encrypt data of size lower
than the RSA modulus n, usually around 1024 bits (64 bits for this exercise!).

We usually encrypt messages with a private encryption scheme like AES-256 with
a symmetric key k. The key k of fixed size 256 bits for example is then exchanged
via the aymmetric RSA.
-}

--------------------------------------------------------------------------------
-- PART 1 : asymmetric encryption

-- | Returns the greatest common divisor of its two arguments
gcd :: Int -> Int -> Int
gcd m n
    | n == 0    = m
    | otherwise = gcd n (mod m n)

-- | Euler Totient function
phi :: Int -> Int
phi m = length [i | i <- [1..m], gcd m i == 1] -- list comprehension

{-|
Calculates (u, v, d) the gcd (d) and Bezout coefficients (u and v)
such that au + bv = d
-}
computeCoeffs :: Int -> Int -> (Int, Int)
computeCoeffs a 0 = (1, 0)
computeCoeffs a b = (v, u - q*v)
    where
    (q, r) = quotRem a b
    (u, v) = computeCoeffs b r

-- | Inverse of a modulo m
inverse :: Int -> Int -> Int
inverse a m
    | gcd a m /= 1 = undefined
    | otherwise = u `mod` m
    where u = fst (computeCoeffs a m)

-- | Calculates (a^k mod m)
modPow :: Int -> Int -> Int -> Int
modPow a 0 m = 1 `mod` m -- base case : a^0 mod m = 1
modPow 0 k m = 0         -- base case : 0^k mod m = 0
modPow a k m
    | even k    = x
    | otherwise = (a * x) `mod` m
    where
    j = k `div` 2
    x = modPow ((a^2) `mod` m) j m

-- | Returns the smallest integer that is coprime with phi
smallestCoPrimeOf :: Int -> Int
smallestCoPrimeOf a = test a 2
```



```haskell
    where
    test :: Int -> Int -> Int
    -- takes two number x and y
    -- returns the smallest Co-prime of x, by checking the candidate numbers from y
    test x y
        | gcd x y == 1 = y
        | otherwise    = test x (y+1)


{-|
Generates keys pairs (public, private) = ((e, n), (d, n))
given two "large" distinct primes, p and q
-}
genKeys :: Int -> Int -> ((Int, Int), (Int, Int))
genKeys p q = ((e, n), (d, n))
    where
    d = inverse e x
    x = (p-1)*(q-1)
    n = p*q
    e = smallestCoPrimeOf x


-- | This function performs RSA encryption
rsaEncrypt :: Int        -- ^ value to encrypt
           -> (Int, Int) -- ^ public key
           -> Int
rsaEncrypt x (e, n) = modPow x e n

-- | This function performs RSA decryption
rsaDecrypt :: Int        -- ^ value to decrypt
           -> (Int, Int) -- ^ public key
           -> Int
rsaDecrypt c (d, n) = modPow c d n

--------------------------------------------------------------------------------
-- PART 2 : symmetric encryption

-- | Returns position of a letter in the alphabet
toInt :: Char -> Int
toInt x
    | isAsciiLower x = ord x - ord 'a'
    | isAsciiUpper x = ord x - ord 'A'
    | otherwise = undefined

-- | Returns the n^th letter
toChar :: Int -> Char
toChar n = chr (n + ord 'a')

-- | "adds" two letters
add :: Char -> Char -> Char
add c1 c2 = toChar ((toInt c1 + toInt c2) `mod` 26)

-- | "subtracts" two letters
subtract :: Char -> Char -> Char
subtract c1 c2
    | result < 0 = toChar (result + 26)
    | otherwise = toChar result
    where result = toInt c1 - toInt c2

-- the next functions present
-- 2 modes of operation for block ciphers : ECB and CBC
-- based on a symmetric encryption function e/d such as "add"

-- | ecb (electronic codebook) encryption with block size of a letter
ecbEncrypt :: Char -> [Char] -> [Char]
```

```
132: ecbEncrypt k m = map addK m
133:     where
134:         addK :: Char -> Char
135:         addK c1 = add c1 k
136:
137:
138: -- | ecb (electronic codebook) decryption with a block size of a letter
139: ecbDecrypt :: Char -> [Char] -> [Char]
140: ecbDecrypt k m = map subK m
141:     where
142:         subK :: Char -> Char
143:         subK c1 = subtract c1 k
144:
145: -- | cbc (cipherblock chaining) encryption with block size of a letter
146: cbcEncrypt :: Char   -- ^ public key
147:            -> Char   -- ^ initialisation vector 'iv'
148:            -> [Char] -- ^ message 'm'
149:            -> [Char]
150: -- cbcEncrypt k iv m = loopC (l - 1)
151: cbcEncrypt k iv "" = [] -- base case when the message is empty
152: -- Recursive case: When there's a message, calculate the next ciphertext block 
'c'
153: -- and continue with the rest of the message.
154: cbcEncrypt k iv m  = c : cbcEncrypt k c (tail m)
155:     where
156:         -- 1st version
157:         -- l = length m
158:         -- addK :: Char -> Char
159:         -- addK y = add y k
160:         -- loopC :: Int -> [Char]
161:         -- loopC 0 = [addK (add (head m) iv)]   -- creating c1 (basecase)
162:         -- loopC i = loopC (i-1) ++ [addK (add (m!!i) (loopC (i-1) !! (i-1)))]
163:         c = head (ecbEncrypt (add k iv) m)
164:         -- Calculate the next ciphertext block 'c' by applying ECB encryption 
with a modified IV.
165:
166:
167: -- | cbc (cipherblock chaining) decryption with block size of a letter
168: cbcDecrypt :: Char   -- ^ private key
169:            -> Char   -- ^ initialisation vector 'iv'
170:            -> [Char] -- ^ message 'm'
171:            -> [Char]
172: cbcDecrypt k iv "" = [] -- base case
173: cbcDecrypt k iv c = x : cbcDecrypt k (head c) (tail c)
174:     where
175:         x = subtract (head (ecbDecrypt k c)) iv
176:
177:
178:
179:
```

```
    1: -------- Test Output --------
    2: copying crypto.cabal from skeleton
    3: Resolving dependencies...
    4: Build profile: -w ghc-9.2.8 -O1
    5: In order, the following will be built (use -v for more details):
    6:  - crypto-0.1.0.0 (lib) (first run)
    7:  - crypto-0.1.0.0 (test:crypto-test) (first run)
    8:  - crypto-0.1.0.0 (test:crypto-properties) (first run)
    9: Configuring library for crypto-0.1.0.0..
   10: Preprocessing library for crypto-0.1.0.0..
   11: Building library for crypto-0.1.0.0..
   12: [1 of 1] Compiling Crypto           ( src/Crypto.hs, /tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/build/Crypto.o, ⬀
/tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/build/Crypto.dyn_o )
   13: Configuring test suite 'crypto-test' for crypto-0.1.0.0..
   14: Configuring test suite 'crypto-properties' for crypto-0.1.0.0..
   15: Preprocessing test suite 'crypto-properties' for crypto-0.1.0.0..
   16: Building test suite 'crypto-properties' for crypto-0.1.0.0..
   17: Preprocessing test suite 'crypto-test' for crypto-0.1.0.0..
   18: Building test suite 'crypto-test' for crypto-0.1.0.0..
   19: [1 of 1] Compiling Main             ( test/Props.hs, ⬀
/tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-properties/build/crypto-properties/crypto-properties-tmp/Main.o )
   20: [1 of 1] Compiling Main             ( test/Tests.hs, ⬀
/tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-test/build/crypto-test/crypto-test-tmp/Main.o )
   21: Linking /tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-properties/build/crypto-properties/crypto-properties ...
   22: Linking /tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-test/build/crypto-test/crypto-test ...
   23: Resolving dependencies...
   24: Build profile: -w ghc-9.2.8 -O1
   25: In order, the following will be built (use -v for more details):
   26:  - crypto-0.1.0.0 (lib) (configuration changed)
   27:  - crypto-0.1.0.0 (test:crypto-test) (configuration changed)
   28: Configuring library for crypto-0.1.0.0..
   29: Preprocessing library for crypto-0.1.0.0..
   30: Building library for crypto-0.1.0.0..
   31: Configuring test suite 'crypto-test' for crypto-0.1.0.0..
   32: Preprocessing test suite 'crypto-test' for crypto-0.1.0.0..
   33: Building test suite 'crypto-test' for crypto-0.1.0.0..
   34: Running 1 test suites...
   35: Test suite crypto-test: RUNNING...
   36: Test suite crypto-test: PASS
   37: Test suite logged to:
   38: /tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-test/test/crypto-0.1.0.0-crypto-test.log
   39: 1 of 1 test suites (1 of 1 test cases) passed.
   40: copying test from skeleton
   41: Resolving dependencies...
   42: Build profile: -w ghc-9.2.8 -O1
   43: In order, the following will be built (use -v for more details):
   44:  - crypto-0.1.0.0 (lib) (configuration changed)
   45:  - crypto-0.1.0.0 (test:crypto-test) (configuration changed)
   46:  - crypto-0.1.0.0 (test:crypto-properties) (dependency rebuilt)
   47: Configuring library for crypto-0.1.0.0..
   48: Preprocessing library for crypto-0.1.0.0..
   49: Building library for crypto-0.1.0.0..
   50: Configuring test suite 'crypto-test' for crypto-0.1.0.0..
   51: Preprocessing test suite 'crypto-properties' for crypto-0.1.0.0..
   52: Building test suite 'crypto-properties' for crypto-0.1.0.0..
   53: [1 of 1] Compiling Main             ( test/Props.hs, ⬀
/tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-properties/build/crypto-properties/crypto-properties-tmp/Main.o )
   54: Linking /tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-properties/build/crypto-properties/crypto-properties ...
   55: Preprocessing test suite 'crypto-test' for crypto-0.1.0.0..
   56: Building test suite 'crypto-test' for crypto-0.1.0.0..
   57: [1 of 1] Compiling Main             ( test/Tests.hs, ⬀
```

```
/tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-test/build/crypto-test/crypto-test-tmp/Main.o )
   58: Linking /tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-test/build/crypto-test/crypto-test ...
   59: Resolving dependencies...
   60: Build profile: -w ghc-9.2.8 -O1
   61: In order, the following will be built (use -v for more details):
   62:  - crypto-0.1.0.0 (lib) (configuration changed)
   63:  - crypto-0.1.0.0 (test:crypto-test) (configuration changed)
   64: Configuring library for crypto-0.1.0.0..
   65: Preprocessing library for crypto-0.1.0.0..
   66: Building library for crypto-0.1.0.0..
   67: Configuring test suite 'crypto-test' for crypto-0.1.0.0..
   68: Preprocessing test suite 'crypto-test' for crypto-0.1.0.0..
   69: Building test suite 'crypto-test' for crypto-0.1.0.0..
   70: Running 1 test suites...
   71: Test suite crypto-test: RUNNING...
   72: Test suite crypto-test: PASS
   73: Test suite logged to:
   74: /tmp/d20231013-38-dge43q/dist-newstyle/build/x86_64-linux/ghc-9.2.8/crypto-0.1.0.0/t/crypto-test/test/crypto-0.1.0.0-crypto-test.log
   75: 1 of 1 test suites (1 of 1 test cases) passed.
   76:
   77: -------- Test Errors --------
   78: Checking https://repo1.maven.org/maven2/org/scala-lang/scala3-library_3/
   79: Checked https://repo1.maven.org/maven2/org/scala-lang/scala3-library_3/
   80: Downloading https://repo1.maven.org/maven2/org/scala-lang/scala3-library_3/
   81: Downloaded https://repo1.maven.org/maven2/org/scala-lang/scala3-library_3/
   82: Checking https://repo1.maven.org/maven2/org/scala-lang/scala3-library_3/maven-metadata.xml
   83: Checked https://repo1.maven.org/maven2/org/scala-lang/scala3-library_3/maven-metadata.xml
   84: Downloading https://repo1.maven.org/maven2/org/scala-lang/scala3-library_3/maven-metadata.xml
   85: Downloaded https://repo1.maven.org/maven2/org/scala-lang/scala3-library_3/maven-metadata.xml
   86: Warning: The package list for 'hackage.haskell.org' is 44 days old.
   87: Run 'cabal update' to get the latest list of available packages.
   88: Warning: The package list for 'hackage.haskell.org' is 44 days old.
   89: Run 'cabal update' to get the latest list of available packages.
   90: Warning: The package list for 'hackage.haskell.org' is 44 days old.
   91: Run 'cabal update' to get the latest list of available packages.
   92: Warning: The package list for 'hackage.haskell.org' is 44 days old.
   93: Run 'cabal update' to get the latest list of available packages.
```