PART B: CRYPTOGRAPHY

1. Why are prime numbers important in cryptography?

Cryptography extensively works with numbers, and almost all integers are made of prime numbers. Also, one of the widely used encryption algorithms – RSA is based on primes. The main idea is that we need a mechanism that makes it easy to encrypt the message, but very hard to decipher. Primes offer such a mechanism: it is very easy to multiply two large prime numbers and get a "public key" (used to cipher the message), but it is very hard to get these primes from the key itself (this problem is called - "prime factorization"), therefore, it is very hard to decrypt.

2. Are 150 and 175 co-prime?

Two primes A and B are co-prime if the only common factor is 1. So to answer this question, we start by identifying the factors of each number:

- Factors of 150 are 1, 2, 3, 5, 6, 10, 15, 25, 30, 50, 75 and 150.
- Factors of 175 are 1, 5, 7, 25, 35, 175.

As we can see, both numbers are dividable by 1, 5, and 25, therefore we can say that 150 and 175 are NOT CO-PRIME.

3. State Euler's Phi function and show the result of applying Euler's Phi function to the number 10.

The Euler's Phi function $\Phi(N)$ measures the divisibility of the number. It shows how many integers are less than or equal to N that do not share any common factors (greater than one) with N. Let's apply this function to the number 10 and check the basic idea:

- We start by preparing the list of integers from 1 to N: [1,2,3,4,5,6,7,8,9,10]
- Now let us write down all factors of number 10: (1, 2, 5, 10)
- Now we remove from the initial list all numbers that share the 2 as the factor:
- [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- Same for the factor of 5: [1, 3, 5, 7, 9] and 10: [1, 3, 7, 9]
- We count the length of the list and write it down as the result: $\Phi(10) = 4$

Counting the Euler's Phi function is time-consuming, but hopefully, this is not the case for the prime numbers. The Phi function for any prime number N is equal to N-1.

4. Explain why Euler's Phi function is important for cryptography

As I stated before RSA algorithm is very important to modern cryptography. It is easy to use, provides great safety for the data transferred, and is hard to crack. RSA is a public-key cryptosystem, as it is based on the idea that every user has a "public key" that he

provides openly for anyone who wants to contact him. He also has a “private key” that he keeps secret and uses to decipher messages addressed to him using his public key. So the basic need of such an algorithm is the function that can provide a simple and fast way to “lock” a message using a public key, while making it hard to “unlock” it with a public key, but without the private key. These keys work as “inverts” of each other. The idea of such a function is called a “one-way function” - it is easy to compute in one way (to get the cyphered message from the original one) but hard to invert (to get the initial message from the cyphered one). The Euler's Phi function is an example of such a “one-way” function: it is easy to compute the public key – we multiply two large prime numbers, while it is very hard to get the private key – get these two numbers just from the product number. The fact that the Euler's Phi function is basically a “locking mechanism” of the most popular modern encryption algorithm makes it very important.

5. Write a pseudo-code algorithm that implements the modulo function

```
function mod(p, q):
    // Handle the wrong argument
    if q == 0:
        print("Error! Division by zero!")
        return

    // When both have the same sign or p is zero
    if (p*q) >= 0:
        // No iterations needed
        if (p >= 0 and p < q) or (p < 0 and p > q) or (p == 0):
            return p
        // Recursive call after argument adjustment
        else:
            return mod(p-q, q)

    // When both have different sign
    else:
        // Adjust the argument
        return mod(p+q, q)
```

6. **Alice is sending a message to Bob. Bob wants Alice to encode the message using RSA. Bob thinks that the following elements are necessary in order for Alice to encode the message:**
   - two prime numbers multiplied together to make N
   - a value called e which is co-prime with phi(N)
   6.1. State the formula that will allow Alice to take message M and encrypt it to E using e and N

E = (M^e)mod(N)

6.2. Given an N of 21, select a value for e. Show your working. Do you agree that this forms Bob's public key?

A value of "e" should be co-prime with Φ(N), so we start by calculating it. N = 21, but we can write it as a product of 2 primes: 3 * 7 = 21. So, using the multiplicative property of the Euler's phi function for relatively prime numbers, and the fact that phi value for any prime N Φ(N) = N-1, we get: Φ(21) = Φ(3) * Φ(7) = (3-1) * (7-1) = 2 * 6 = 12. Now we can select any number that is co-prime with "12" as the value for "e". Factors of 12 are: 1 * (2^2) * 3, so we can select "e" = 17, because a number 17 is co-prime with phi(N) = 12 . The pair of values (N, e) or (21, 17) can form Bob's public key. The only problem with this key is that the numbers themselves are small, and it makes this key not safe. The basic idea is correct and this key can be used for learning purposes (not the real life encryption).

6.3. Alice wants to send a message which is just a number to test the encryption is working. Choose a small number for Alice's message (M<20) and encrypt it with Bob's public key. Show your working. Make it clear what your chosen M is.

I decided to select M = 4. Now, to encrypt this message we use the formula from the 6.1 and values from 6.2: M = 4, N = 21, e = 17. The encrypted message will look like this: E = (M^e) mod (N) = (4^17) mod (21) = 17179869184 mod (21) = 16. So using the Bob's public key (21,17) we ciphered original message M = 4 to E = 16.