*Question 1: Classical Cryptography** [25 marks]

1.1 Key size of a substitution cipher and how to break it [5 marks]

- A substitution cipher maps each of the 26 letters to a unique letter. Thus, the number of possible permutations (i.e., key size) is **26! ≈ 2^88.4**.
- Despite the large key space, it can be broken using **frequency analysis**. Since the cipher preserves the letter frequency (e.g., 'E' appears most frequently in English), attackers can map the most frequent ciphertext characters to the expected plaintext letters.

- # Revised Answer for 1.2 Kasiski test for Vigenère cipher key length estimation [5 marks]

  - The **Kasiski test** identifies repeated sequences in the ciphertext and measures the distances between their occurrences.
  - In the given example, repeated sequences like **"KIOV"** occur at regular intervals (e.g., positions 1 and 10). The distance between these repetitions is 9.
  - To estimate the key length, factorize these distances (e.g., 9), and calculate the **greatest common divisor (GCD)**. The GCD suggests possible key lengths.
  - For example, the GCD of 9 is **3**, which suggests that the key length is likely **3**. The reason is that repeating ciphertext blocks indicate that they are likely encrypted with the same key segment, and thus the key length corresponds to the distance between the repeated patterns.

  By factoring these distances and looking for common factors, we can guess the most probable key lengths.

1.3 Index of Coincidence (IC) and how to use it [10 marks]

- IC measures the probability that two randomly selected letters are the same.
- IC ≈ 0.065 for English text; IC ≈ 0.038 for random text.
- To determine the key length:
  - Try different assumed key lengths (e.g., 1–10).
  - Split ciphertext into `n` substrings (each encrypted with same key letter).
  - Calculate IC for each substring.
  - A key length that yields substring ICs ≈ 0.065 is likely correct.

1.4 Deriving the key given key length [5 marks]

- After identifying the key length (say `k`), split ciphertext into `k` groups.
- Each group is a Caesar cipher, solve each with frequency analysis.
- For each group, identify the shift that aligns most frequent letter with 'E' (or use Chi-square test).
- Combine the `k` shifts to recover the Vigenère key.

Question 2: Public Key Cryptography [25 marks]

2.1 Four security weaknesses in the RSA setup [8 marks]

1. **Modulus generation on host computer**: Private primes `p` and `q` exposed to host.
   → Attacker with host access can factor `N`.
2. **Private key distribution over internal network**:
   → Internal attacker or malware could intercept private keys.
3. **Per-user key pairs generated on computer, not HSM**:
   → Risk of key exposure or weak randomness.
4. **Shared public modulus for all employees**:
   → Vulnerable to **common modulus attacks** if two users share `N` but use different `e`.

2.2 Is HSM private key recovery API secure? [7 marks]

- If HSM can output `d` given `e`, it must store or compute it using `ϕ(N)`.
- Since `ϕ(N)` = `(p−1)(q−1)` and only HSM has `p, q`, this allows HSM to compute `d = e⁻¹ mod ϕ(N)`.
- This is **dangerous**, as anyone can get the private key by querying HSM with `e`.
- Violates basic principles: HSM should never reveal private key.

2.3 Using same RSA key pair for encryption and signing [10 marks]

- RSA Encryption: `C = M^e mod N`, Decryption: `M = C^d mod N`
- RSA Signing: `S = M^d mod N`, Verification: `M = S^e mod N`
- **Problem**: If same key is used:
  - Anyone with `C` can compute `C^d mod N` to get `M` (decryption).
  - Or forge signature by choosing `S`, computing `S^e`, and claiming it's `M`.
- **Textbook RSA is insecure** without padding (e.g., OAEP for encryption, PSS for signing).
- **Conclusion**: Not secure to reuse key pair without padding.