# INFORMATION SECURITY

# LAB MIDTERM

## Name:

M Abdullah Azam

## Reg no:

Sp24-Bse-034

## Submitted to:

Mam Ambreen Gul

## Date

21-10-2025

# Question No # 05:

**Conceptual Des And Aes**

**Part (a):**

**Write one similarity between DES and AES.**

**Answer:**

**Both DES and AES are symmetric key block ciphers.**
**They use the same key for both encryption and decryption.**
**Both operate on fixed-size data blocks to secure information.**


**Part(b):**

**What does CBC mode stand for in block ciphers?**

**Answer:**

**CBC stands for Cipher Block Chaining mode.**
**In CBC, each plaintext block is XORed with the previous ciphertext block.**
**This makes encryption more secure by adding dependency between blocks.**


**Part (c):**

**Why is AES faster than DES?**

**Answer:**

**AES is faster because it uses fewer, simpler rounds than DES.**
**Its algorithm is optimized for modern computer hardware.**
**AES also handles larger block sizes efficiently, improving speed.**

-------------------------------------

# Question No # 02:

Write a Python program to decrypt a message that was encrypted using the Caesar Cipher. The program should take ciphertext (LXFOPVEFRNHR) and key (5) as input and display the plaintext.

Example:

Enter ciphertext: khoor

Enter shift: 3

Plaintext: hello

Answer:

```
C: > Users > HP > Desktop > DEVELOPMENT FILES >  Q2is.py > ...
  1    ciphertext = input("Enter ciphertext: ")
  2    shift = int(input("Enter shift: "))
  3
  4    plaintext = ""
  5
  6    for char in ciphertext:
  7        if char.isalpha():
  8            base = ord('A') if char.isupper() else ord('a')
  9            plaintext += chr((ord(char) - base - shift) % 26 + base)
 10        else:
 11            plaintext += char
 12
 13    print("Plaintext:", plaintext)
```

## Output:

```
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/HP/Desktop/DEVELOPMENT FILES/Q2is.py"
Enter ciphertext: Khoor
Enter shift: 3
Plaintext: Hello
PS C:\Users\HP>
```

## Question NO #03:

Write a Python program to decrypt a ciphertext using the Vigenère Cipher. Ask the user for ciphertext and key, and display the decrypted plaintext.

**Example:**

**Enter ciphertext: LXFOPVEFRNHR**

**Enter key: LEMON**

**Plaintext: ATTACKATDAWN**

**Answer:**

```
> Users > HP > Desktop > DEVELOPMENT FILES > 🐍 Q3is.py > ...
1    ciphertext = input("Enter ciphertext: ").upper()
2    key = input("Enter key: ").upper()
3
4    plaintext = ""
5    key_index = 0
6
7    for char in ciphertext:
8        if char.isalpha():
9            shift = ord(key[key_index % len(key)]) - ord('A')
10           decrypted_char = chr((ord(char) - ord('A') - shift) % 26 + ord('A'))
11           plaintext += decrypted_char
12           key_index += 1
13       else:
14           plaintext += char
15
16   print("Plaintext:", plaintext)
```

**Output:**

```
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/HP/Desktop/DEVELOPMENT FILES/Q3is.py"
Enter ciphertext: LXFOPVEFRNHR
Enter key: LEMON
Plaintext: ATTACKATDAWN
PS C:\Users\HP>
```

# Question No # 04:

**Answer:**

**Code Error:**

**result += chr(ord(char) + shift)**

**This line doesn't wrap alphabets properly.**

**Fixed Code:**

```
> Users > HP > Desktop > DEVELOPMENT FILES >  Q4is.py > ...
1    def caesar_encrypt(text, shift):
2        result = ""
3        for char in text:
4            if char.isalpha():
5                base = ord('A') if char.isupper() else ord('a')
6                result += chr((ord(char) - base + shift) % 26 + base)
7            else:
8                result += char
9        return result
0
1    msg = input("Enter message: ")
2    s = int(input("Enter shift: "))
3    print("Ciphertext:", caesar_encrypt(msg, s))
4    |
```

**Output:**

```
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/HP/Desktop/DEVELOPMENT FILES/Q4is.py"
Enter message:  HELLO
Enter shift: 3
Ciphertext:  KHOOR
PS C:\Users\HP> |
```

Question NO #01:

Write a Python program that performs both encryption and decryption using XOR operation.

Requirements:

1. Ask the user for message and a single-character key.

2. Encrypt the message using XOR (ord() and chr()).

3. Decrypt it by applying XOR again with the same key.

4. Show both ciphertext and decrypted plaintext.

Answer:

```
> Users > HP > Desktop > DEVELOPMENT FILES > 🐍 Q1is.py > ...
1    message = input("Enter message: ")
2    key = input("Enter single-character key: ")
3
4    |
5    ciphertext = ""
6    for char in message:
7        ciphertext += chr(ord(char) ^ ord(key))
8
9    print("Ciphertext:", ciphertext)
10
11
12   decrypted = ""
13   for char in ciphertext:
14       decrypted += chr(ord(char) ^ ord(key))
15
16   print("Decrypted text:", decrypted)
```

Output:

```
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Programs/Python/Python312/python.exe  c:/Users/HP/Desktop/DEVELOPMENT FILES/Q1is.py
Enter message: HELLO
Enter single-character key: A
 Ciphertext:    ♦
Decrypted text: HELLO
PS C:\Users\HP>
```

-------------------------------------------------------------