Date __/__/__
Page ______

4. Write a program for error detecting code using CRC - CCITT.

Sol.
```
def calculate_crc(data: bytes, polynomial:
        int = 0x1021, initial_value : int = 0xFFFF):

    crc = initial_value
    for byte in data:
        crc ^= byte << 8

        for _ in range (8):
            if (crc & 0x8000)
                crc = (crc << 1) ^ polynomial
            else:
                crc <<= 1

        crc &= 0xFFFF
    return crc.

~~dep~~ ~~def~~
def append_crc (data: bytes) → bytes:
    crc = calculate_crc ~~modified~~ (data)
    return data + crc.to_bytes (2, byteorder = 'big')

def verify_crc (data_with_crc : bytes) → bool:
    if len(data_with_crc) < 2:
        return False
    data = data_with_crc [: -2]
    recieved = int.from_bytes (data_with_crc[-2:],
                        byteorder = 'big')
    calculated = calculate_crc (data)
    return recieved == calculated
```

```
message = b"Hello, CRC!"

data = append-crc (message)
print(f"Data with crc: {data-with-crc.hex()}")

is-valid = verify-crc (data)
print(f"Is the CRC valid? {is-valid}")

corrupted = data[:-1] + bytes([data[-1] ^ 0xFF])
print(f"Corrupted data: {corrupted.hex()}")

is-valid = verify-crc(corrupted)
print(f"Is the CRC valid for corrupted data?
        {is-valid}")
```

[illegible]

[illegible]

[illegible]
26/12/24