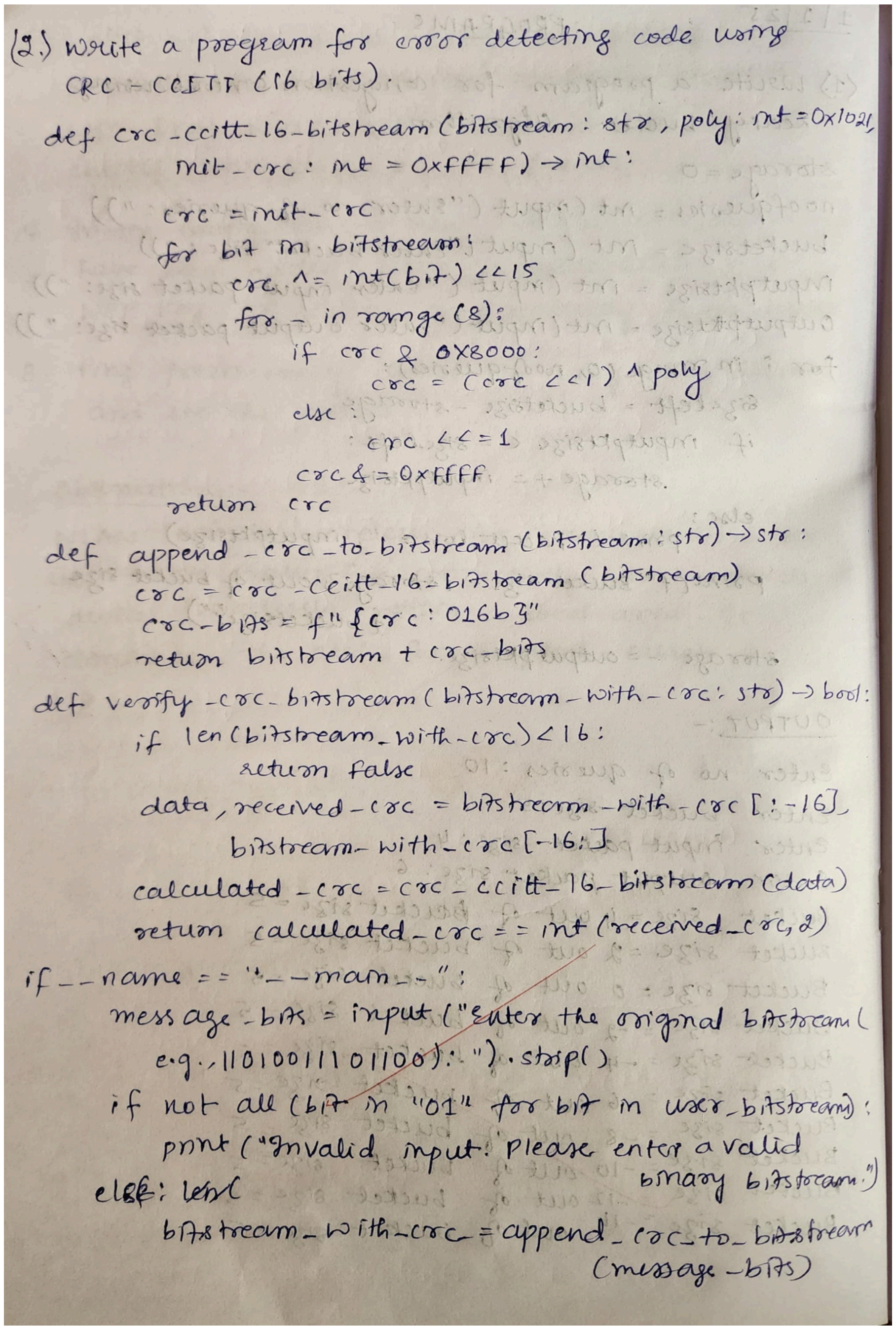
1/1/25 Experiment-13

Program 1:

Question:

Write a program for error detecting code using CRC-CCITT (16-bits).



(2.) write a program for error detecting code using CRC-CCITT (16 bits).

```
def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021,
        init_crc: int = 0xFFFF) -> int:
    crc = init_crc
    for bit in bitstream:
        crc ^= int(bit) << 15
        for _ in range(8):
            if crc & 0X8000:
                crc = (crc << 1) ^ poly
            else:
                crc <<= 1
            crc &= 0xFFFF
    return crc
def append_crc_to_bitstream(bitstream: str) -> str:
    crc = crc_ccitt_16_bitstream(bitstream)
    crc_bits = f"{crc:016b}"
    return bitstream + crc_bits
def verify_crc_bitstream(bitstream_with_crc: str) -> bool:
    if len(bitstream_with_crc) < 16:
        return False
    data, received_crc = bitstream_with_crc[:-16],
        bitstream_with_crc[-16:]
    calculated_crc = crc_ccitt_16_bitstream(data)
    return calculated_crc == int(received_crc, 2)
if __name__ == "__main__":
    message_bits = input("Enter the original bitstream (
        e.g., 1101001110 1100): ").strip()
    if not all(bit in "01" for bit in user_bitstream):
        print("Invalid input: Please enter a valid
                                binary bitstream.")
    else:
        bitstream_with_crc = append_crc_to_bitstream
                                (message_bits)
```

```
print (f"Transmitted bitstream with CRC:
                {bitstream_with_crc}")
user_bitstream = input("Enter the received
    bitstream for verification: ").strip()
if not all (bit in "01" for bit in user_bitstream)
    print("Invalid input. Please enter a valid
            binary bitstream").
elif len (user_bitstream) < 16:
    print("Invalid input. Received bitstream
        must include at least 16 bits for CRC.")
else:
    is_valid = verify_crc_bitstream (user_
                                    bitstream)
    if is_valid:
        print("No errors detected. CRC Valid")
    else:
        print ("Error detected! CRC invalid")
```

OUTPUT:

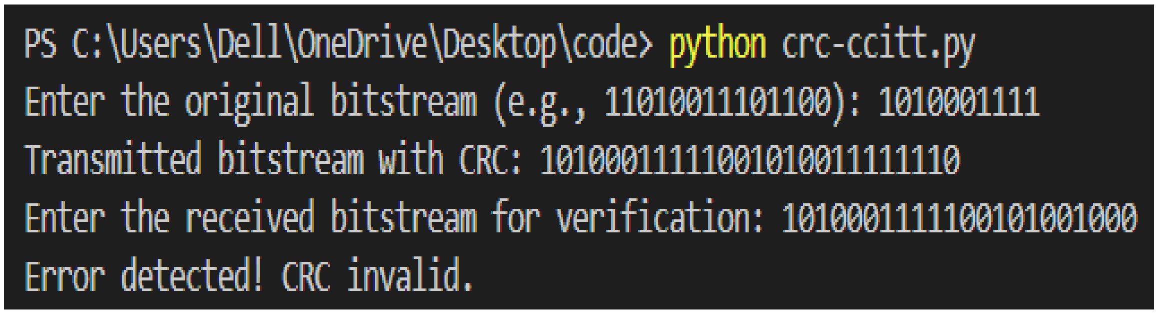
```
Enter the original bitstream : 1010001111
Transmitted bitstream with CRC : 1010001111 0010100
                                          1111110
Enter the received bitstream for verification:
                1010011111006101001000
Error detected! CRC invalid.
```

Code:

```python
def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021, init_crc: int = 0xFFFF) -> int:
    crc = init_crc
    for bit in bitstream:
        crc ^= int(bit) << 15
        for _ in range(8):
            if crc & 0x8000:
                crc = (crc << 1) ^ poly
            else:
                crc <<= 1
            crc &= 0xFFFF
    return crc

def append_crc_to_bitstream(bitstream: str) -> str:
    crc = crc_ccitt_16_bitstream(bitstream)
    crc_bits = f"{crc:016b}"
    return bitstream + crc_bits

def verify_crc_bitstream(bitstream_with_crc: str) -> bool:
    if len(bitstream_with_crc) < 16:
        return False
    data, received_crc = bitstream_with_crc[:-16], bitstream_with_crc[-16:]
    calculated_crc = crc_ccitt_16_bitstream(data)
    return calculated_crc == int(received_crc, 2)

if __name__ == "__main__":
    message_bits = input("Enter the original bitstream (e.g., 11010011101100): ").strip()
    if not all(bit in "01" for bit in message_bits):
        print("Invalid input. Please enter a binary bitstream (e.g., 11010011101100).")
    else:
        bitstream_with_crc = append_crc_to_bitstream(message_bits)
        print(f"Transmitted bitstream with CRC: {bitstream_with_crc}")
        user_bitstream = input("Enter the received bitstream for verification: ").strip()
        if not all(bit in "01" for bit in user_bitstream):
```

```python
        print("Invalid input. Please enter a valid binary bitstream.")
    elif len(user_bitstream) < 16:
        print("Invalid input. Received bitstream must include at least 16 bits for CRC.")
    else:
        is_valid = verify_crc_bitstream(user_bitstream)
        if is_valid:
            print("No errors detected. CRC valid.")
        else:
            print("Error detected! CRC invalid.")
```

Output:



```
PS C:\Users\Dell\OneDrive\Desktop\code> python crc-ccitt.py
Enter the original bitstream (e.g., 11010011101100): 1010001111
Transmitted bitstream with CRC: 10100011111001010011111110
Enter the received bitstream for verification: 1010001111100101001000
Error detected! CRC invalid.
```