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 wring
   CRC - CCITT (16 bits).
 def crc -ccitt-16-bitstream (bitstream: 8+2, poly: nt = 0x102)
         mit-crc: mt = Oxffff) -> mt:
         crevemit-crowner) tuged the some
        for bit in bitstream; tugin ) in spressions
    in tobo cre 1= int(bit) (215)
   in rouge (8): grajona - mattatuque
                 if coc & 0x8000;
                     cac = (cac 201) Apoly
                    if inputphisize d=34 ons
                 Crc4= 0xFFFF - - - Doorte.
         return crc
 def append -crc-to-bitstream (bitstream: str) -> str:
       crc = crc -ceitt-16=bitstream (bitstream).
       crc-biAs = f" fcrc: 016b3"
       return bitstream + coc-bits goods
 def verify -crc-birstream (birstream-with-crc: sto) -> bool:
       if len (bitstream_with_core) < 16:
             return false OI: 1286 DUD TO ON 50 FO
       data, received - exc = bitstream - with - exc [:-16]
             bitstream-with-crof-16: Jana
       calculated - crc = crc - ccitt-16-bitstrocom (doda)
       return calculated_crc == int (received_crc, 2)
if -- name == "-- man -- ";
     message - bits = input ("Enter the original bitstorand
        e.g., 11010011101100): "). stap()
      if not all (bit in "ot" for bit in war bitstream):
         print ("Invalid input! Please enter a valid
     electi lend bistocami)
         bAs tream_with_crc=append_cocto_bistream
                                   (message -bits)
```

point (f" Transmitted bitstream with CRC: {bitotocom-with-coc3") user - bistream = input ("enter the received bitstream for verification: "). Strip() if not all Chit in "OI" for but in war-bastream print ("Invalid mout Please enter a valid brany bitstream"). elif len (users-bitstream) < 16: point ('Invalid input . Received bitstream must melude at least 16 bAs for CRC.") che: is-valid = verify-crc-bitstream (userif is valid: a contence of business if pmt ("No error detected CFC valid") print ("Error detected! cfc mvalid") else: chentelocket, clasel OUTPUT: Enter the original bitstream: 1010001111 13410393 Transmitted bastream with CRC: 10100011111 0010100 1.00,4611 - supplies 1010

Enter the received bits bream for verification:

Error detected! CRC invalid.

Code:

```
def crc ccitt 16 bitstream(bitstream: str, poly: int = 0x1021, init crc: int = 0xFFFF) -> int:
  crc = init crc
  for bit in bitstream:
     \operatorname{crc} \stackrel{\wedge}{=} \operatorname{int}(\operatorname{bit}) << 15
     for in range(8):
        if crc & 0x8000:
          crc = (crc << 1) \land poly
       else:
          crc <<= 1
        crc &= 0xFFFF
  return crc
def append crc to bitstream(bitstream: str) -> str:
  crc = crc ccitt 16 bitstream(bitstream)
  ere 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):
```

```
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.")</pre>
```

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.