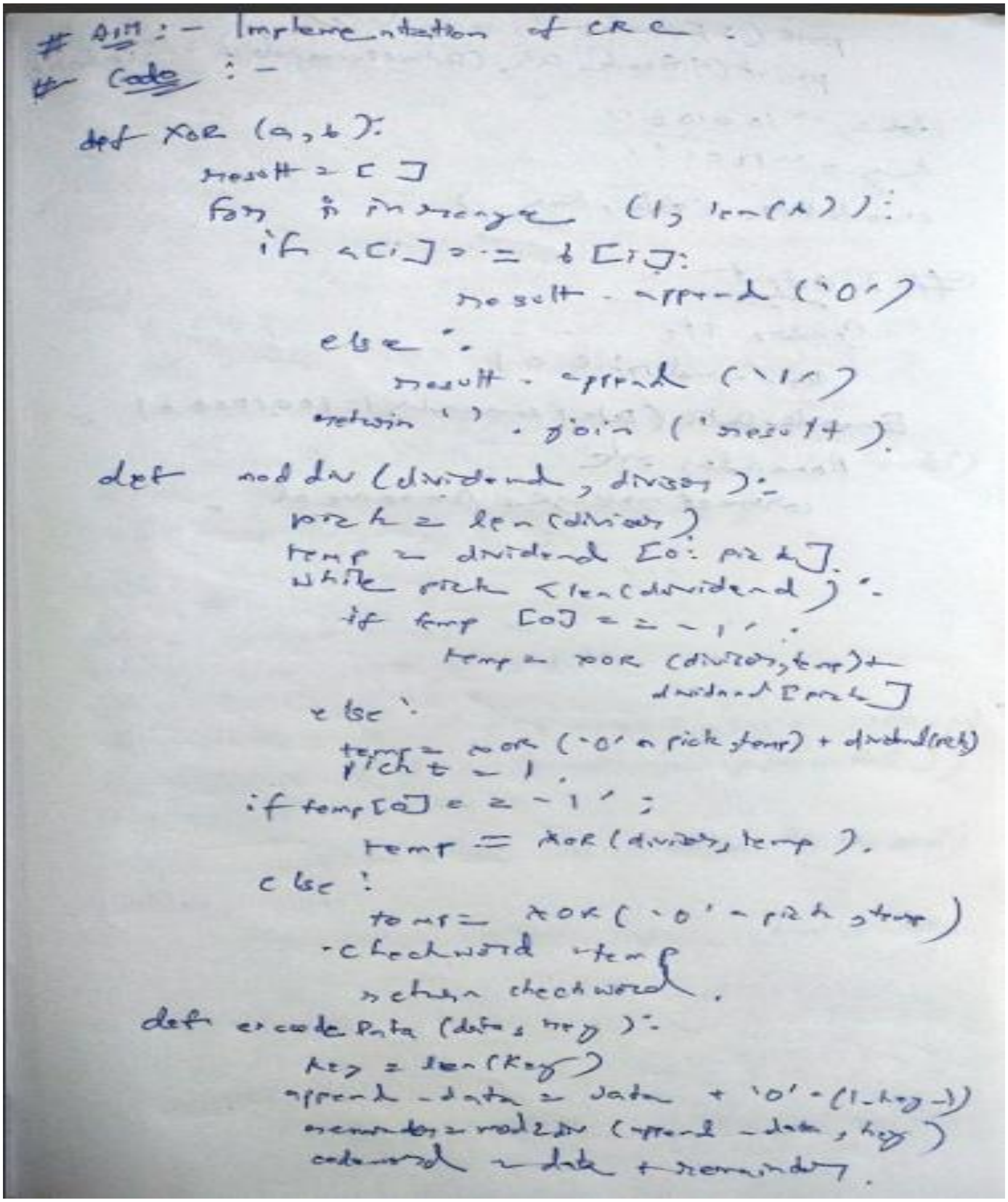
# PART-B

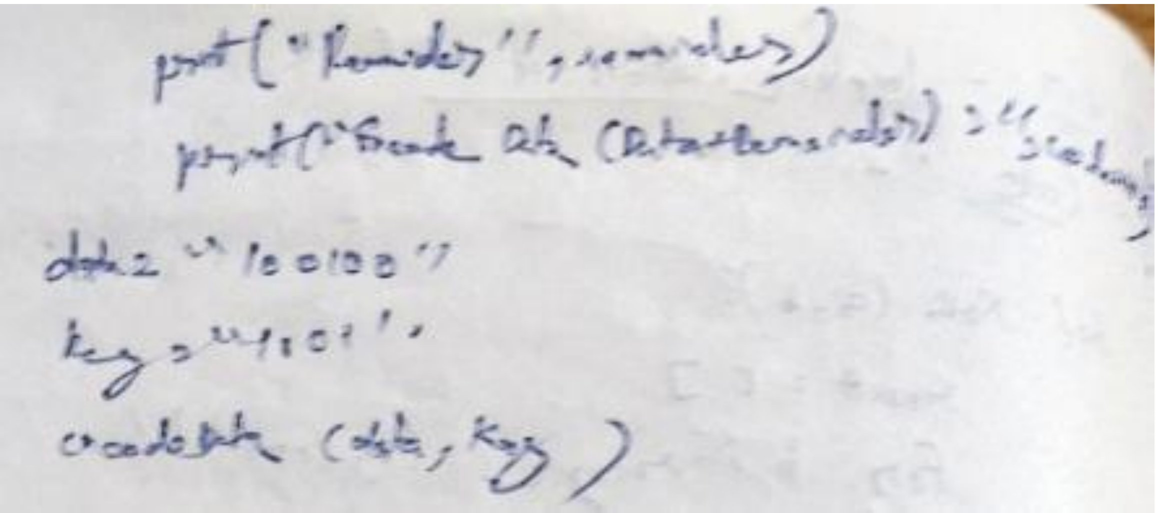
**<u>Program 14</u>**
Write a program for error detecting code using CRC-CCITT (16-bits).

**Code :**

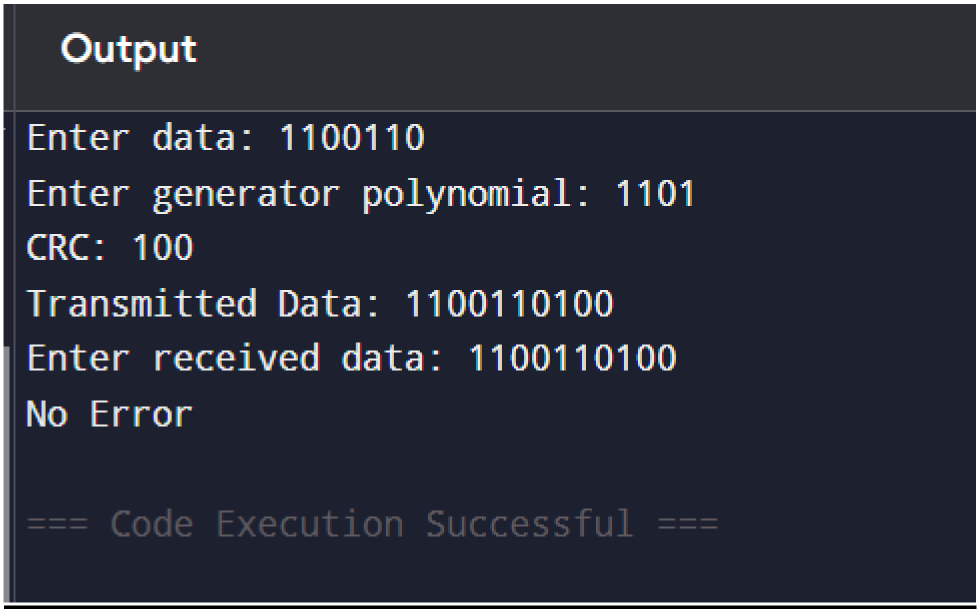


```
# AIM :- Implementation of CRC :
# Code :-

def XOR (a, b):
        result = [ ]
        for i in range (1, len(b)):
            if a[i] == b[i]:
                    result . append ('0')
            else :
                  result . append ('1')
            return ''. join (result).
  def   mod div (dividend, divisor):-
            pick = len (divisor)
            temp = dividend [0: pick].
            while pick < len(dividend) :-
               if temp [0] == '1' :
                    temp = xor (divisor, temp)+
                               dividend [pick]
            else :
               temp = xor ('0' * pick, temp) + dividend[pick]
               pick += 1 .
         if temp [0] == '1' ;
               temp = xor (divisor, temp ).
         else :
               temp = xor ('0' * pick, temp )
            checkword = temp
               return checkword .
def encodeData (data, key ):-
        key = len (key)
        append_data = data + '0' * (l_key -1)
        remainder = mod2div (append_data, key )
        codeword = data + remainder .
```



```
    print("Remainder", remainder)
      print("Encoded Data (Data+Remainder)", codeword
          )
data = "100100"
key = "1101"
encodeData (data, key )
```



```
Enter data: 1100110
Enter generator polynomial: 1101
CRC: 100
Transmitted Data: 1100110100
Enter received data: 1100110100
No Error

=== Code Execution Successful ===
```