Name : Saakshi Navale

CRC (Cyclic Redundancy Check) Cycle II Labs

Sender side & Reciever side

```
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 mod2div (divident, divisor):
    pick = len (divisor)
    tmp = divident [0 : pick]
    while pick < len (divident):
        if tmp[0] == '1':
            tmp = xor (divisor, tmp) + divident [pick]

        else:
                         '0' * pick
            tmp = xor (~~'0' * pick~~ ~~divisor~~, tmp) + divident [pick]


        pick += 1

    if tmp[0] == '1':
        tmp = xor (divisor, tmp)

    else:
        tmp = xor ('0' * pick , tmp)
    checkword = tmp
    return checkword
```

```
def   encodedata (data, key):
      l-key = len (key)
      appended_data = data + '0' * (l-key - 1)
      remainder = mod2 div (appended_data, key)

      codeword = data + remainder.
      return codeword


def   decode Data ( data, key):
      l_key = len (key)
      appended_data = data + '0' * (l_key - 1)
      remainder = mod2 div (appended_data, key)

      rethen remainder.


def   CRC :

      data = input (" Enter data u want to send:");
      key = "1001"
   encodedans = encodeData (data, key)
      print (ans);

      // if u want make change to ans.

      decodedans = decodeData (encodedans, key)

      checkzero = "0" * (len(key) - 1)

      if (ans == tmp):
            print (" The transfer is without error");

      else:
            print (" Packet Not transferred due
                   to Error Detection");
```