1. **CRC CODE**

```
#crc client
import socket

def comp(dw,g):
    dw_padded= dw + '0'*(len(g)-1)
    dw_list=list(map(int,dw_padded))
    g_l=list(map(int,g))
#performing crc xor

    for i in range(len(dw)):
        if dw_l[i] == 1:
            for j in range(len(g)):
                dw_l[i+j]^=g_l[j]

#cnvert remainder
    rem= ''.join(map(str,dw_l[-(len(g)-1):]))
    return rem

def main():

    c=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    addr=('localhost',12345)
    print('cncting to {} port{}'.format(*addr))
    c.connect(addr)

    try:
        dw=input("get dw:")
        g=input("enter gen:")
        rem = comp(dw,g)
        cw=dw+rem
        c.sendall(cw.encode())
        c.sendall(g.encode())
    finally:
        c.close()
if __name__ == '__main__':
    main()

#------------------------------------------------------------------------
#crc server

import socket

def comp(codeword, generator):
    rem=""
    cw_l=list(map(int,cw))
    g_l=list(map(int,g))

    for i in range(len(cw_-len(g)+1):
        if cw_l[i] == 1:
            for j in range(len(g)):
                cw_l[i+j]^=g_l[j]

    rem = ''.join(map(str,cw_l[-(len(g)-1):]))
    return rem

def main():

    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    addr = ('localhost',12345)
    print('strtng {}on port{}'.format(*addr))
    s.bind(addr)

    s.listen(1)

    while True:
        print('waiting for cnct')
        c,a = s.accept()

        try:
            print('cnction from',a)
            cw = c.recv(1024).decode()
            print('cw =',cw)

            g = c.recv(1024).decode()
            print('gen =',g)

            rem=comp(cw,g)
            if int(rem)==0:
                print('crct code')
            else:
                print('error ')
        finally:
            c.close()
if __name__ =='__main__':
    main()
```

2. HAMMING CODE

```
#server
import socket

def calc_redundant_bits(m):
    r = 0
    while 2 ** r < m + r + 1:
        r += 1
    return r

def pos_redundant_bits(data, r):
    m = len(data)
    res = ''
    j = 0
    k = 0

    for i in range(1, m + r + 1):
        if i == 2 ** j:
            res += '0'
            j += 1
        else:
            res += data[-1 - k]
            k += 1

    return res[::-1]

def calc_parity_bits(arr, r):
    n = len(arr)
    for i in range(r):
        val = 0
        for j in range(1, n + 1):
            if j & (2 ** i) == (2 ** i):
                val ^= int(arr[-j])
        arr = arr[:n - (2 ** i)] + str(val) + arr[n - (2 ** i) + 1:]
    return arr

def detect_error(arr, r):
    n = len(arr)
    res = 0
    for i in range(r):
        val = 0
        for j in range(1, n + 1):
            if j & (2 ** i) == (2 ** i):
                val ^= int(arr[-j])
        res += val * (10 ** i)
    return int(str(res), 2)

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('localhost', 1234))
s.listen(5)
print("Server is connected..")

c_soc, addr = s.accept()
print("Connected with ", addr)

while True:
    choice = "\nEnter your choice: \n 1. Convert to hamming code \n 2. Check for error"
    c_soc.send(choice.encode())

    ch = c_soc.recv(1024).decode()

    if ch == '1':
        c_soc.sendall(("Enter the code:").encode())
        data = c_soc.recv(1024).decode()
        m = len(data)
        r = calc_redundant_bits(m)
        arr = pos_redundant_bits(data, r)
        arr = calc_parity_bits(arr, r)
        print("Hamming code: ", arr)
        c_soc.sendall(arr.encode())

    elif ch == '2':
        c_soc.sendall(("Enter the code: ").encode())
        arr = c_soc.recv(1024).decode()
        m = len(arr)
        r = calc_redundant_bits(m)
        correction = detect_error(arr, r)
        if correction == 0:
            c_soc.sendall("There is no error!".encode())
        else:
            print('crction :',correction)
            msg = "Error at position " + str(len(arr) - correction + 1) + " from left i.e bit = " +
str(correction) +" "
            c_soc.sendall(msg.encode())
    else:
        c_soc.sendall("Enter a valid choice!".encode())

#client
#client
```

```
import socket

c = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
c.connect(('localhost',1234))
print ("Connected to server!")

while True:
    msg = c.recv(1024).decode()
    print(msg)
    ch = input("Enter your choice: ")

    if ch=='1':
        c.sendall(ch.encode())

        msg = c.recv(1024).decode()
        print(msg)
        data = input()
        c.sendall(data.encode())

        hamming_code = c.recv(1024).decode()
        print("hamming code: " , hamming_code)

    elif ch=='2':
         c.sendall(ch.encode())

         msg = c.recv(1024).decode()
         print(msg)
         data = input()
         c.sendall(data.encode())

         msg = c.recv(1024).decode()
         print(msg)
```