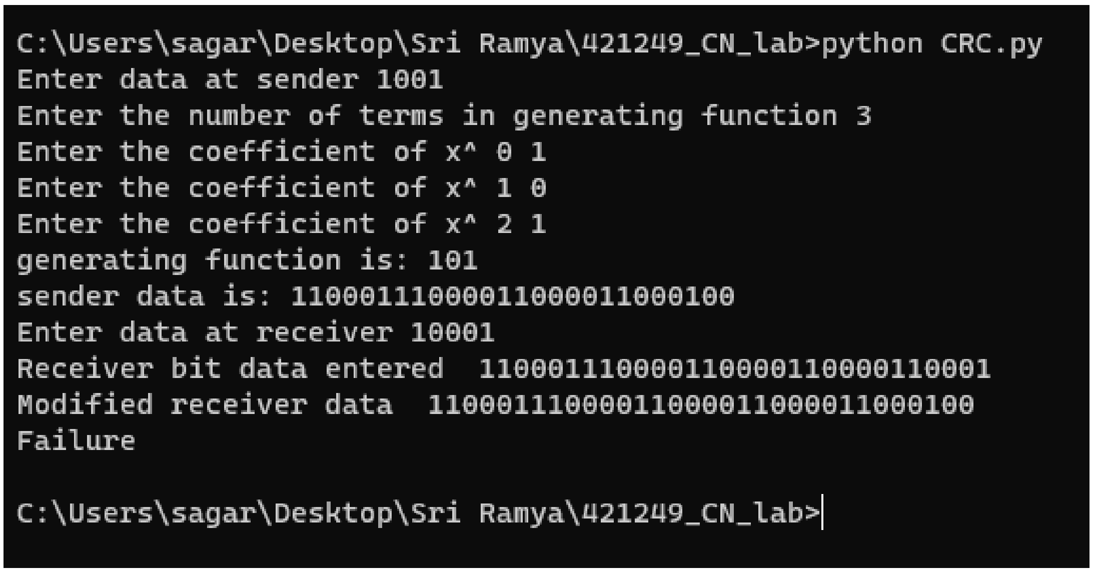
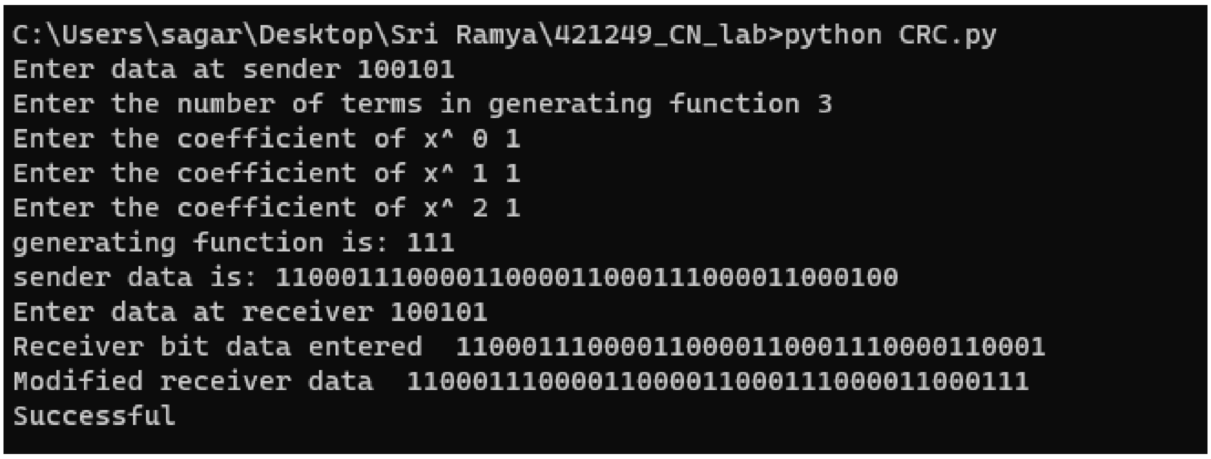
Cyclic Redundancy Check

```
def byte_data(a):
    ans=""
    for i in a:
        ans+=bin(ord(i))[2:]
    return ans

def division(data):
    while(len(data)>=terms):
        xor=data[0:terms]
        ans=int(xor,2) ^ int(gen_fun,2)
        if(ans!=0):
            ans=bin(ans)[2:]
            data=ans+data[terms:]
        else:
            data=data[terms:]
    return data
sender=input("Enter data at sender ")
sender_data=byte_data(sender)
terms=int(input("Enter the number of terms in generating function "))
gen_fun=""
for i in range(terms):
    coeff=input(f"Enter the coefficient of x^ {i} ")
    gen_fun=coeff+gen_fun
print(f'generating function is: {gen_fun}')
for i in range(terms-1):
    sender_data+='0'
print(f'sender data is: {sender_data}')

rems=division(sender_data)
while(rems!="" and len(rems)<terms-1):
    rems='0'+rems
receiver=input("Enter data at receiver ")
receiver_data=byte_data(receiver)
print("Receiver bit data entered ",receiver_data)
receiver_data=receiver_data+rems
print("Modified receiver data ",receiver_data)
remr=division(receiver_data)
'''while(remr and remr.startswith('0')):
    if(remr=="0"):
        remr=""
    else:
        remr=remr[1:]'''

if(remr==""):
    print( "Successful ")
else:
    print("Failure ")
```



```
C:\Users\sagar\Desktop\Sri Ramya\421249_CN_lab>python CRC.py
Enter data at sender 1001
Enter the number of terms in generating function 3
Enter the coefficient of x^ 0 1
Enter the coefficient of x^ 1 0
Enter the coefficient of x^ 2 1
generating function is: 101
sender data is: 1100011100001100001100100
Enter data at receiver 10001
Receiver bit data entered  110001110000110000110000110001
Modified receiver data  11000111000011000011000011000100
Failure

C:\Users\sagar\Desktop\Sri Ramya\421249_CN_lab>
```

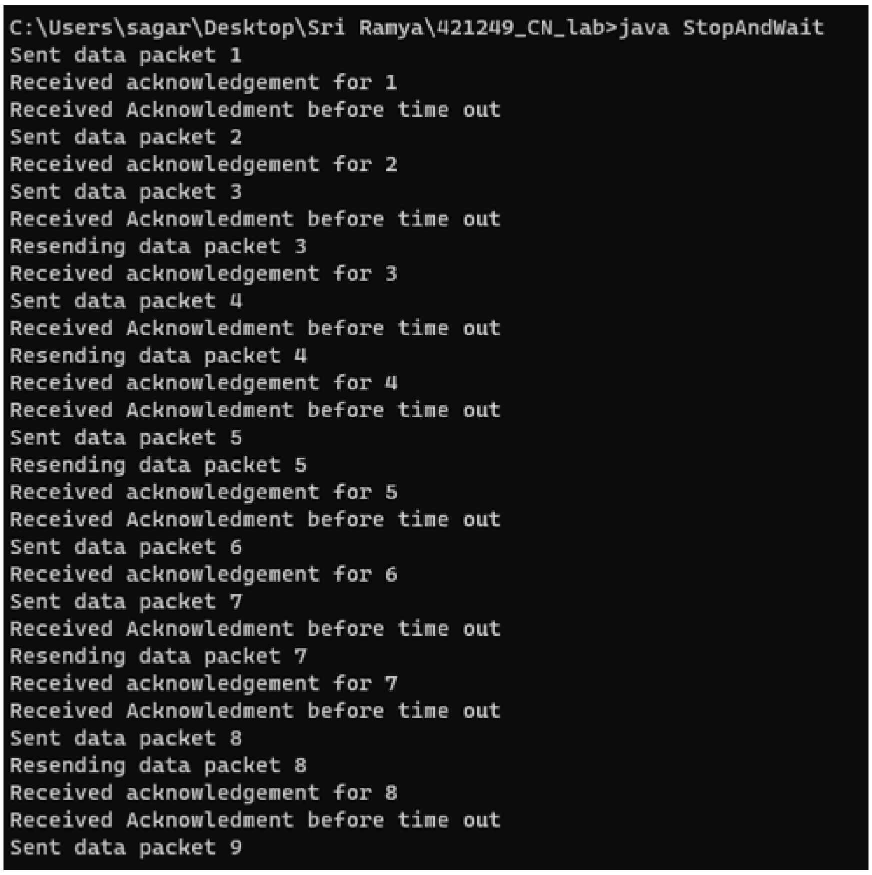


```
C:\Users\sagar\Desktop\Sri Ramya\421249_CN_lab>python CRC.py
Enter data at sender 100101
Enter the number of terms in generating function 3
Enter the coefficient of x^ 0 1
Enter the coefficient of x^ 1 1
Enter the coefficient of x^ 2 1
generating function is: 111
sender data is: 11000111000011000011000111000011000100
Enter data at receiver 100101
Receiver bit data entered  110001110000110000110001110000110001
Modified receiver data  11000111000011000011000111000011000111
Successful
```

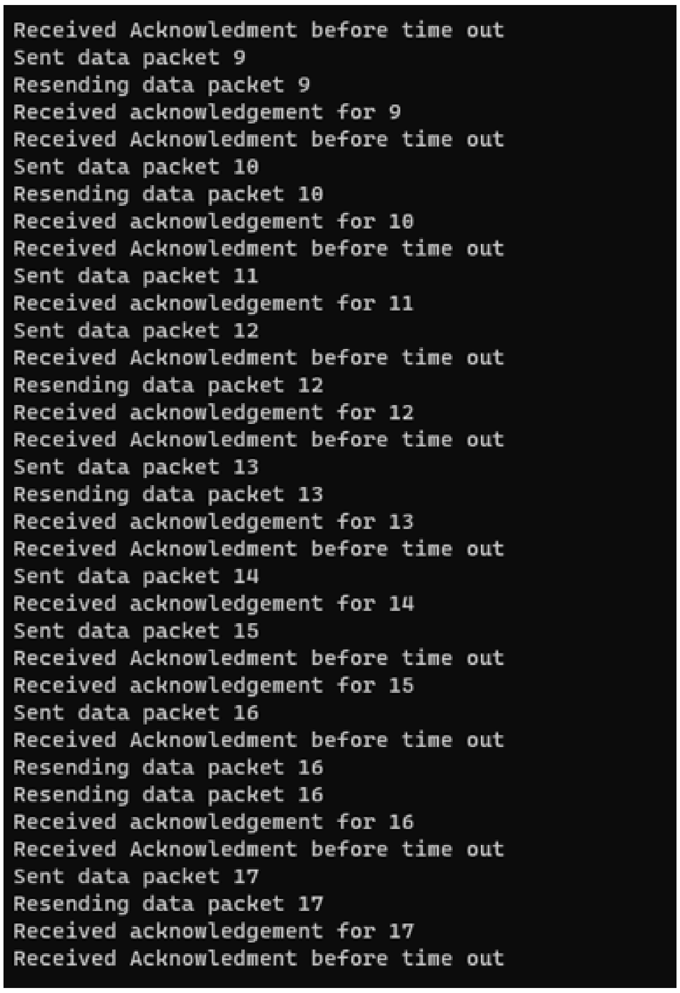
Construct a program for Stop and Wait Protocol. (Example: Sender sends a message and Receiver should give acknowledgement)

```java
import java.util.Random;
class SharedObject {
    volatile int receive=1;
    volatile int flag=0;
    volatile int send=0;
    synchronized public void increment_sender(){
        send++;
        return ;
    }
    synchronized public void increment_receive(){
        receive++;
        return;
    }
}
class Sender implements Runnable{
    SharedObject so;
    public Sender(SharedObject so){
        this.so=so;
    }
    public void run(){

        while(true) {
            synchronized (so) {
                if (so.flag == 0) {
                    so.increment_sender();
                    System.out.println("Sent data packet " + so.send);
                } else {
                    System.out.println("Resending data packet " + so.send);
                }
                so.flag = 1;
            }
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                System.out.println("Received Acknowledment before time out ");

            }
        }
    }
}
class Receiver implements Runnable{
    SharedObject so;
    Thread st;
    public Receiver(SharedObject so,Thread st){
        this.so=so;
        this.st=st;
    }
    public void run(){
        while(true) {

                Random r = new Random();
                int num = r.nextInt(2);
                if (num == 0) {
                    continue;
                } else {
                    int arr[] = {0, 1000};
                    int ind = r.nextInt(2);
                    try {
                        Thread.sleep(arr[ind]);
                    } catch (InterruptedException e) {
                        System.out.println("Receiver thread is interrupted ");
                    }
                    synchronized(so) {
                        if(so.flag==1) {
                            System.out.println("Received acknowledgement for "
+ so.receive);

                            so.flag = 0;
                            so.increment_receive();
                            st.interrupt();//write code to wake the sleeping
sender thread
                        }


                    }
                }
            }
        }
```

```java
}
public class StopAndWait {

    public static void main(String[] args) {
        SharedObject so= new SharedObject();

        Thread s= new Thread(new Sender(so));
        Thread r= new Thread(new Receiver(so,s));
        s.start();
        r.start();

        try{
            s.join();
            r.join();
        }
        catch(InterruptedException e){
            System.out.println("Main thread ends before the other threads join
");
        }

    }
}
```



```
C:\Users\sagar\Desktop\Sri Ramya\421249_CN_lab>java StopAndWait
Sent data packet 1
Received acknowledgement for 1
Received Acknowledment before time out
Sent data packet 2
Received acknowledgement for 2
Sent data packet 3
Received Acknowledment before time out
Resending data packet 3
Received acknowledgement for 3
Sent data packet 4
Received Acknowledment before time out
Resending data packet 4
Received acknowledgement for 4
Received Acknowledment before time out
Sent data packet 5
Resending data packet 5
Received acknowledgement for 5
Received Acknowledment before time out
Sent data packet 6
Received acknowledgement for 6
Sent data packet 7
Received Acknowledment before time out
Resending data packet 7
Received acknowledgement for 7
Received Acknowledment before time out
Sent data packet 8
Resending data packet 8
Received acknowledgement for 8
Received Acknowledment before time out
Sent data packet 9
```



```
Received Acknowledment before time out
Sent data packet 9
Resending data packet 9
Received acknowledgement for 9
Received Acknowledment before time out
Sent data packet 10
Resending data packet 10
Received acknowledgement for 10
Received Acknowledment before time out
Sent data packet 11
Received acknowledgement for 11
Sent data packet 12
Received Acknowledment before time out
Resending data packet 12
Received acknowledgement for 12
Received Acknowledment before time out
Sent data packet 13
Resending data packet 13
Received acknowledgement for 13
Received Acknowledment before time out
Sent data packet 14
Received acknowledgement for 14
Sent data packet 15
Received Acknowledment before time out
Received acknowledgement for 15
Sent data packet 16
Received Acknowledment before time out
Resending data packet 16
Resending data packet 16
Received acknowledgement for 16
Received Acknowledment before time out
Sent data packet 17
Resending data packet 17
Received acknowledgement for 17
Received Acknowledment before time out
```