**Stop and Wait**

import time

a = [i for i in range(10)]

print(a)

b = [None]\*10

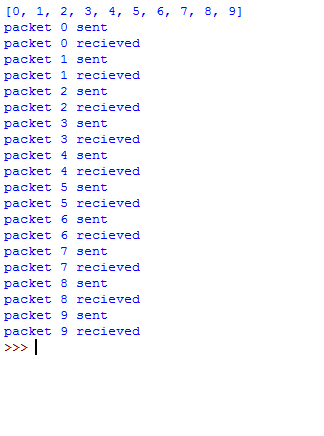
for i in range(10):

b[i] = a[i]

print('packet',i,'sent')

time.sleep(2)

print('packet',i,'recieved')



**Sliding Window**

import time

import random

a = [int(input()) for i in range(9)]

b = [None]\*9

print(a)

window = int(input('enter size of window '))

c = 0

d = 0

for i in range(int(len(a)/window)):

for j in range(window):

b[j] = a[j]

print(c,'th frame sent',)

c+=1

c1 = [d+i for i in range(3)]

print('\n')

t = random.randint(1,5)

while t>2:

t = random.randint(1,5)

time.sleep(t)

print('resending')

time.sleep(t)

if t<=2:

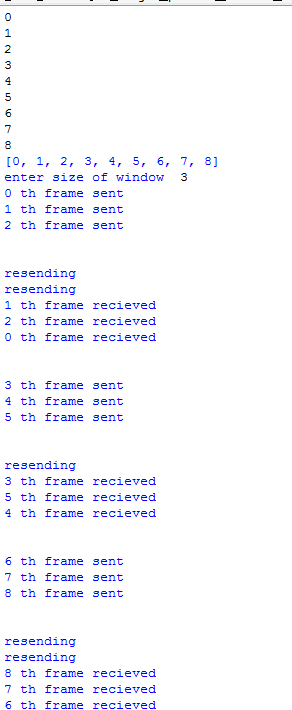
random.shuffle(c1)

for j in range(window):

print(c1[j],'th frame recieved')

d+=1

print('\n')



**Checksum**

def binary(n,m):

p = 0

d = []

for i in range(m-1,-1,-1):

d.append(int(n[i])\*pow(2,p))

p+=1

c = sum(d)

return c

k = int(input('enter number of segments'))

m = int(input('enter size of each segment'))

a = []

for i in range(k):

item = input()

a.append(item)

b = []

for i in range(k):

b.append(binary(a[i],m))

print(b)

w = sum(b)

def ones\_complement(x):

return x ^ ((1 << x.bit\_length()) - 1)

w1 = ones\_complement(w)

print(w1)

print('At reciever side')

e = sum(b) + k

q = ones\_complement(e)

print(q)

print('introducing error')

e = b[0]%m

print(e)

item = list(a[0])

p = 0

for i in item:

if p==e and i=='0':

i='1'

elif p==e and i=='1':

item[e]='0'

print('changed')

p+=1

j = ''

for i in range(m):

j+=item[i]

a[0] = j

b = []

for i in range(k):

print(a[i])

b.append(binary(a[i],m))

print(b)

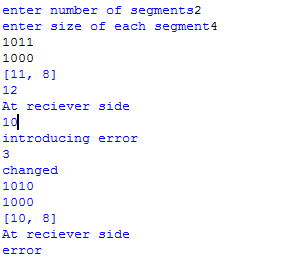
w = sum(b)

print('At reciever side')

e = sum(b) + w1

q = ones\_complement(e)

print(q if q==0 else 'error')



**EVEN ODD PARITY**

e = 0

for i in range(10000):

a = [i%2,0,0,i%2,1,1,0,i%2]

ones = a.count(1)

if ones%2==1:

e+=1

print(e/100)



**Cyclic Redundancy**

def binary\_to\_decimal(n,m):

p = 0

d = []

for i in range(m-1,-1,-1):

d.append(int(n[i])\*pow(2,p))

p+=1

c = sum(d)

return c

m = int(input('enter length of dividend'))

c = int(input('enter length of divisor'))

a = []

b = []

for i in range(m):

item = int(input())

a.append(item)

for i in range(c):

item = int(input())

b.append(item)

for i in range(c-1):

a.append(0)

q = binary\_to\_decimal(a,m)

p1 = binary\_to\_decimal(b,c)

n = q%p1

q = n//2

p = n

d = []

for i in range(c-1):

d.append(p%2)

p = p//2

print(a)

for i in range(1,c):

a[m+c-1-i] = d[i-1]

print('sending to sender ',a)

a = binary\_to\_decimal(a,m+c-1)

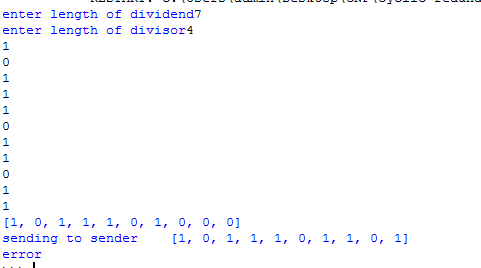
a = a%p1

if a==0:

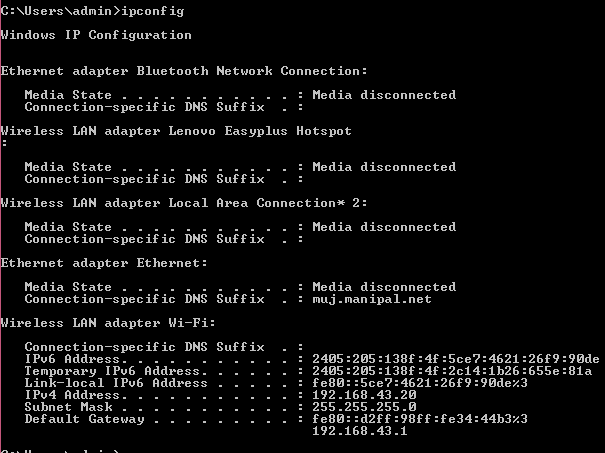
print('no error')

else:

print('error')



**Ipconfig**



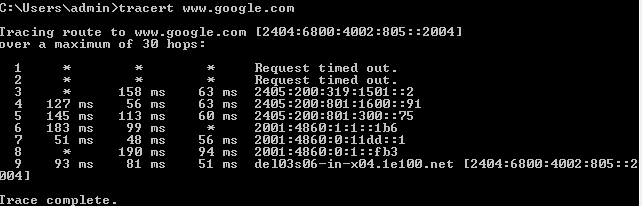
**nslookup**



**route**



**Tracerout**

****