Exercise 1: Framing

Aumrudh Lal Kumar TJ,18BIT034
BTech IT 3rd Year, 5th Sem

1)

Problem: Write a program for counting the character in message in framing of datalink layer

Aim: To write a program which does character count using python

Program:

#charactercountsender

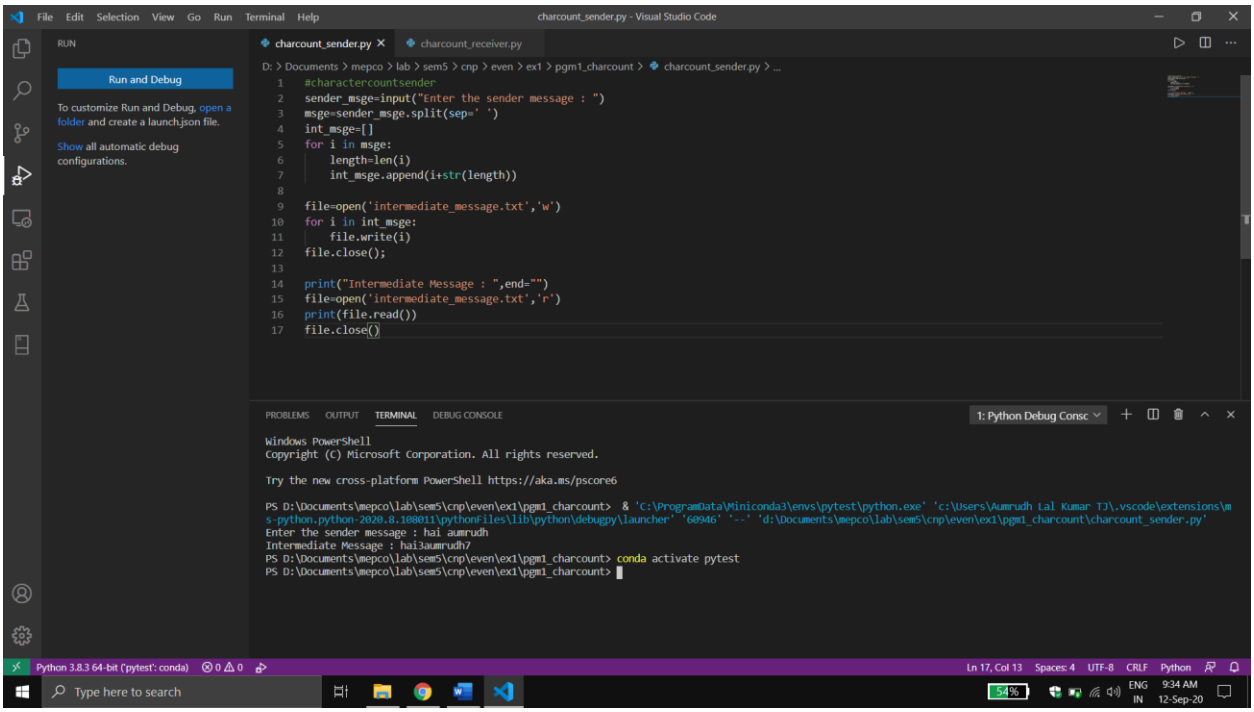
```
sender_msge=input("Enter the sender message : ")
msge=sender_msge.split(sep=' ')
int_msge=[]
for i in msge:
    length=len(i)
    int_msge.append(i+str(length))

#opening a text file and writing
file=open('intermediate_message.txt','w')
for i in int_msge:
    file.write(i)
file.close();

print("Intermediate Message : ",end="")
file=open('intermediate_message.txt','r')
print(file.read())
file.close()
```

#charactercountreceiver

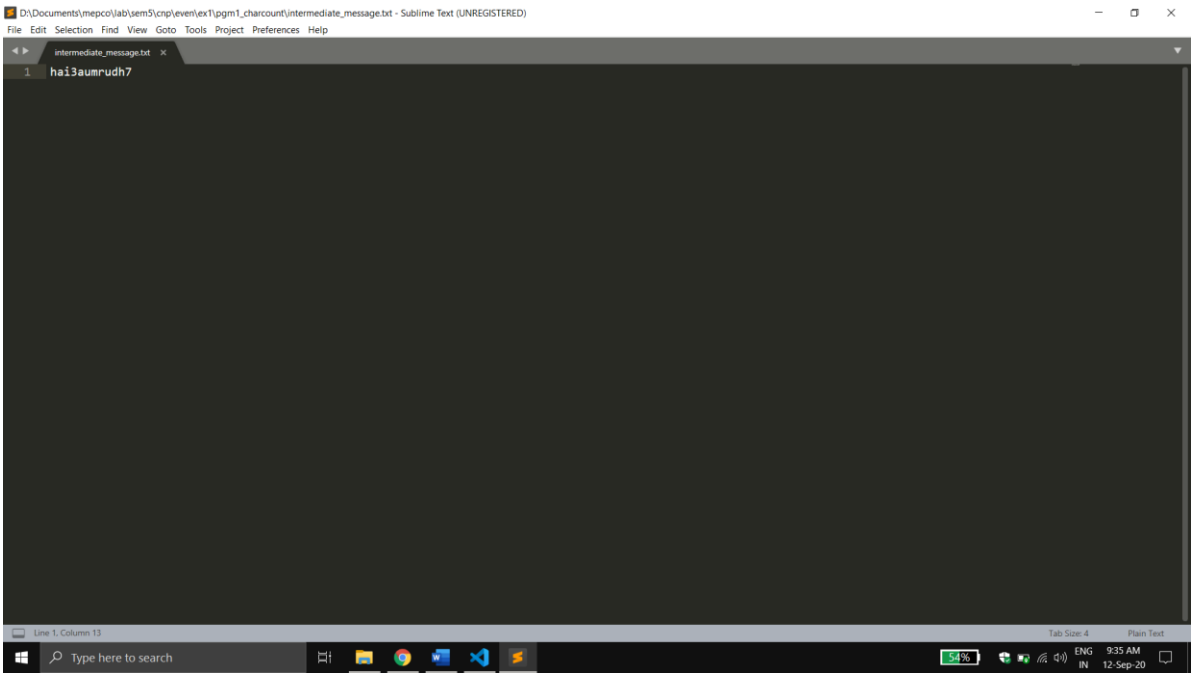
```
#readingfile
File=open('intermediate_message.txt','r')
temp=File.read()
File.close()
t1=len(temp)
i=0
l=[]
flag=0
while(i<t1):
    if(temp[i].isnumeric()):
        x=temp[:i]
        tx=str(len(x))
        if(tx==temp[i]):
            #print("No error")
            l.append((temp[:i]+' '))
        elif tx!=temp[i]:
            print("error")
            flag=1
            break
        #words appended to list are removed from string
        temp=temp[i+1:]
        t1=len(temp)
        i=0
    i+=1
if flag==0:
    for i in l:
        print(i,end='')
```

Output:

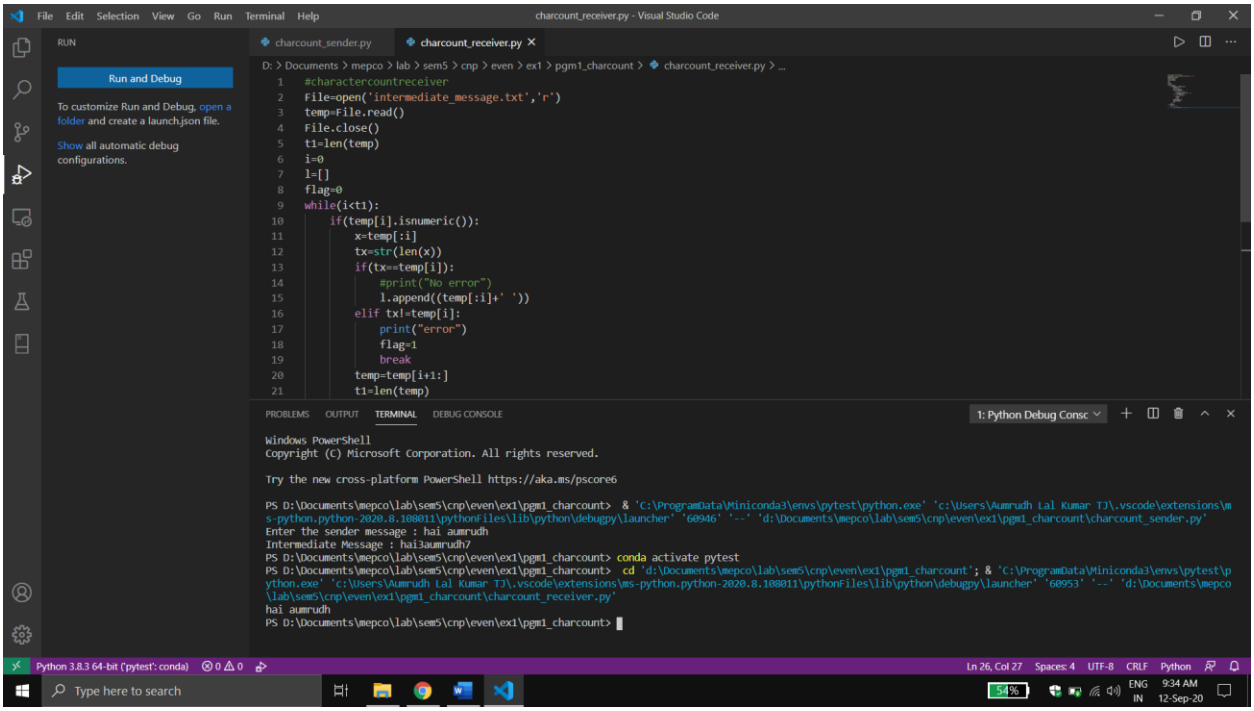
Sender side



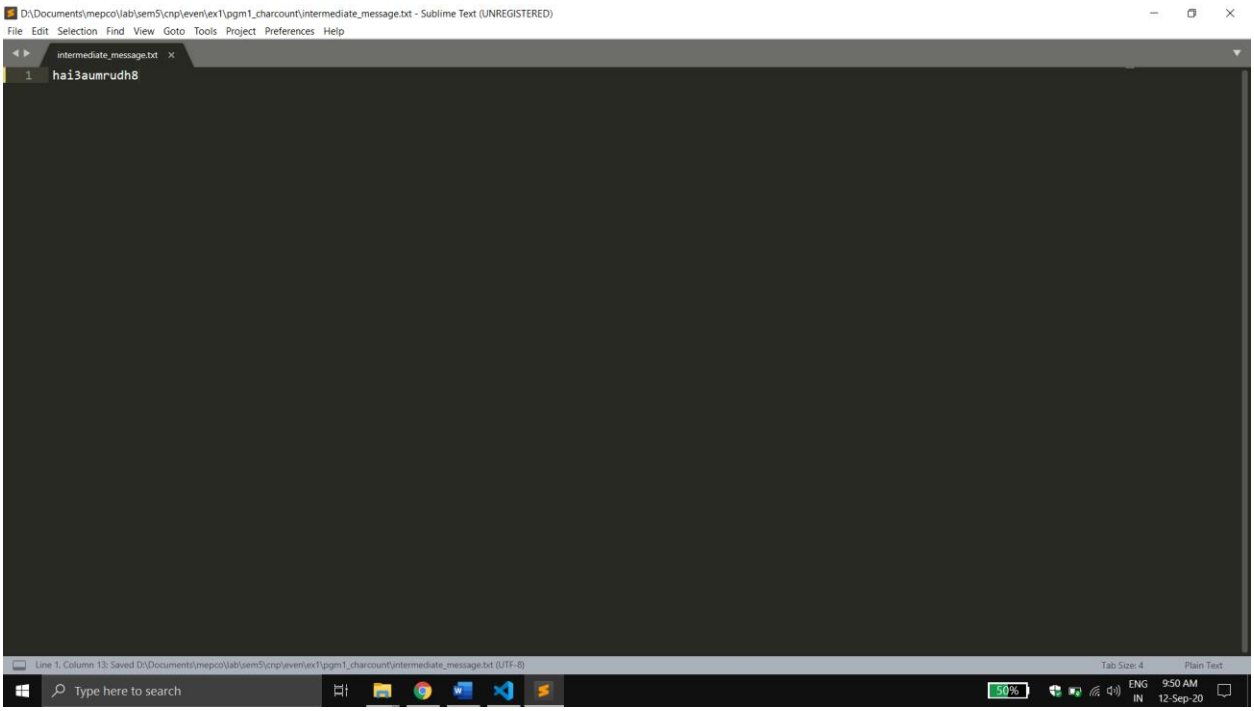
Intermediate text file



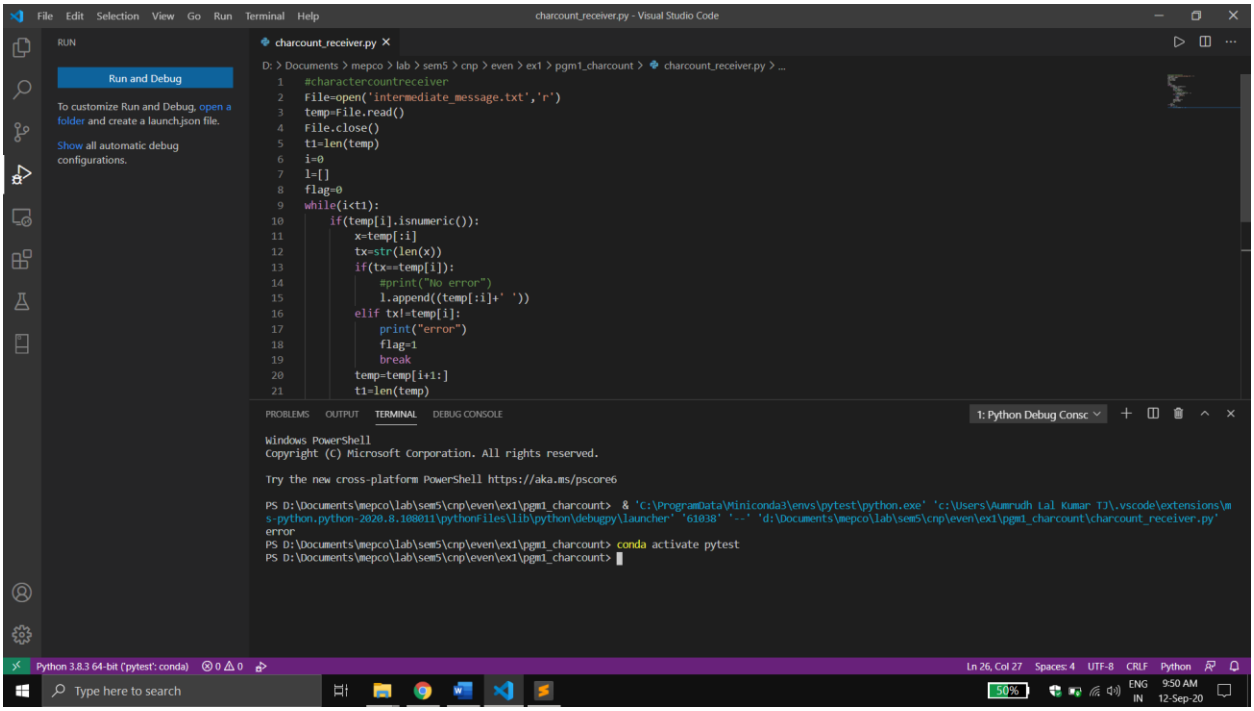
Receiver side



With error: count value changed in text file



Receiver (error)



Result:

The character count done in data link layer using framing was successfully programmed, tested and worked fine in python. The intermediate message was stored with count of character in word after that word. The receiver received the actual message which was sent by sender.

2)

Problem: Write a program for bit stuffing in message in framing of datalink layer

Aim: To write a program which does bit stuffing using python

Program:

#bitstuffsender

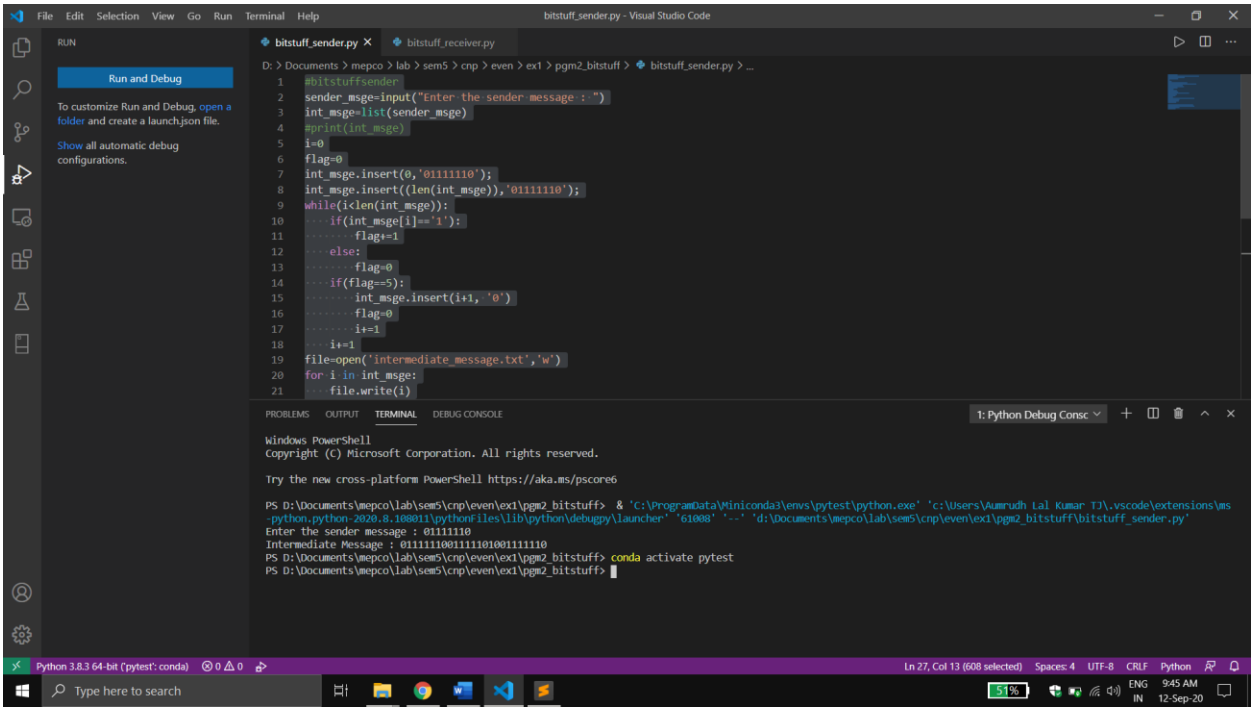
```
sender_msge=input("Enter the sender message : ")
int_msge=list(sender_msge)
#print(int_msge)
i=0
flag=0
int_msge.insert(0,'01111110');
int_msge.insert((len(int_msge)),'01111110');
while(i<len(int_msge)):
    if(int_msge[i]=='1'):
        flag+=1
    else:
        flag=0
    if(flag==5):
        int_msge.insert(i+1, '0')
        flag=0
        i+=1
    i+=1
file=open('intermediate_message.txt','w')
for i in int_msge:
    file.write(i)
file.close();

print("Intermediate Message : ",end="")
file=open('intermediate_message.txt','r')
print(file.read())
file.close()
```

#bitstuffreceiver

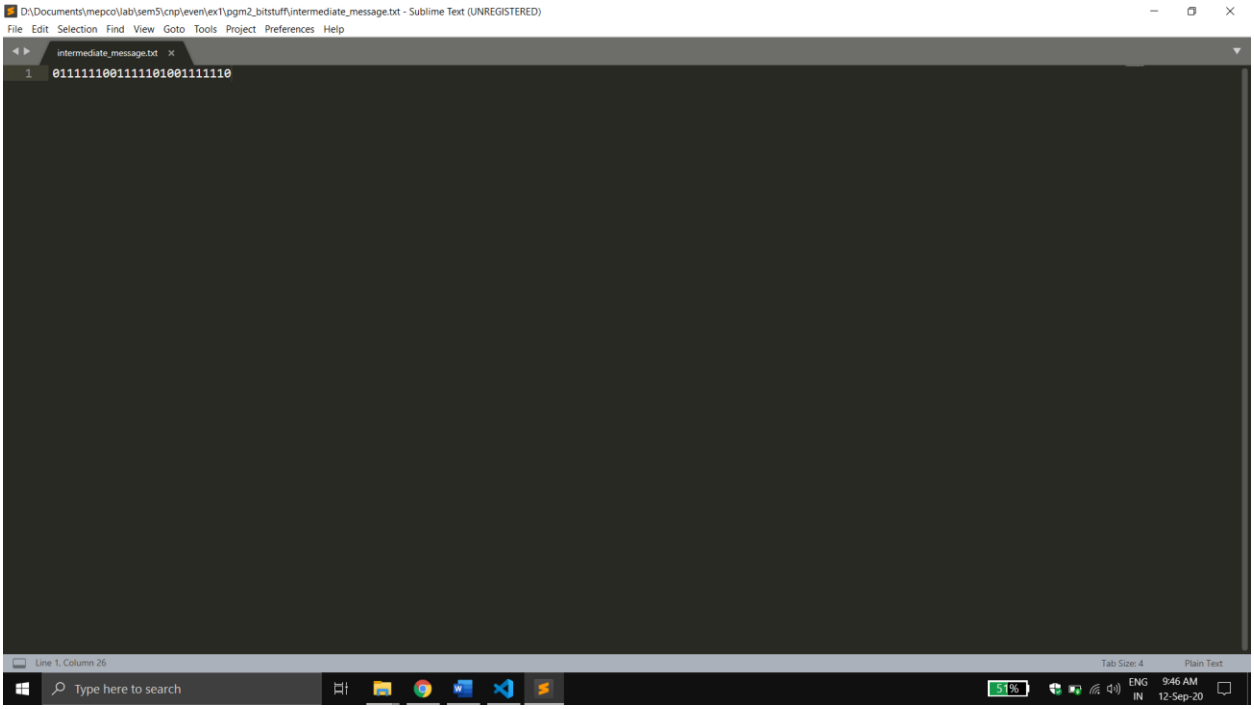
```
File=open('intermediate_message.txt','r')
temp=File.read()
File.close()
#print(temp)
temp1=temp[:8]
#print(temp1)
temp2=temp[-8:]
#print(temp2)
rec_msge=list(temp)
#print(rec_msge)
if(temp1=='01111110' and temp2=='01111110'):
    print("No error")
    t=len(temp)-8
    rec_msge=rec_msge[8:t]
    #print(rec_msge)
    i=0
    flag=0
    print('Receiver Message : ',end='')
    while(i<len(rec_msge)):
        if(rec_msge[i]=='1'):
            flag+=1
        else:
            flag=0
        print(rec_msge[i],end='')
        if(flag==5):
            flag=0
            i+=1
        i+=1
else:
    print('Error')
```

Output:

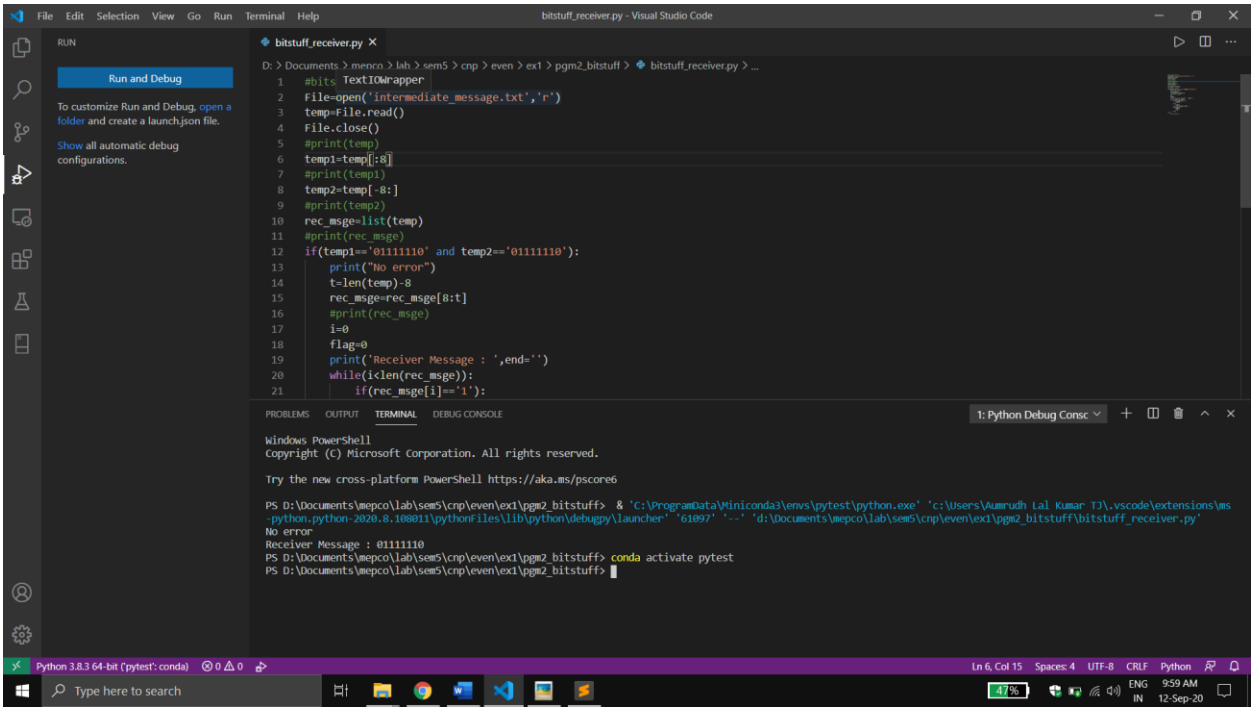
Sender



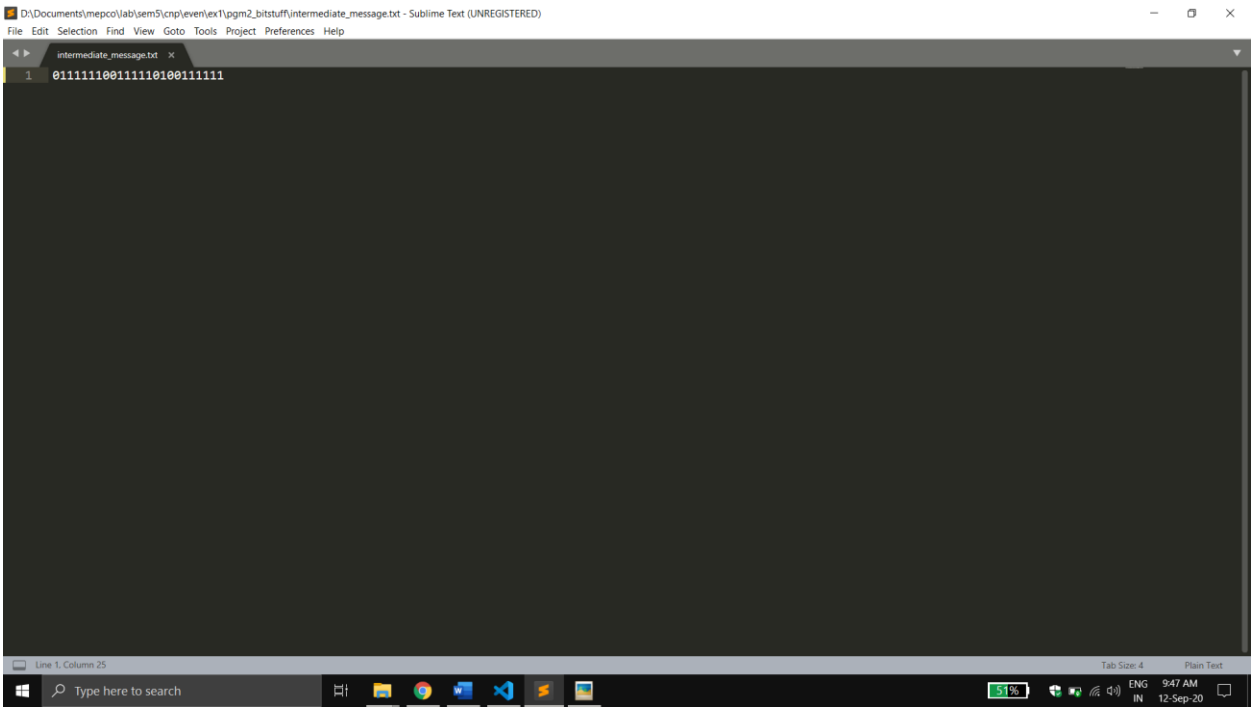
Intermediate text file



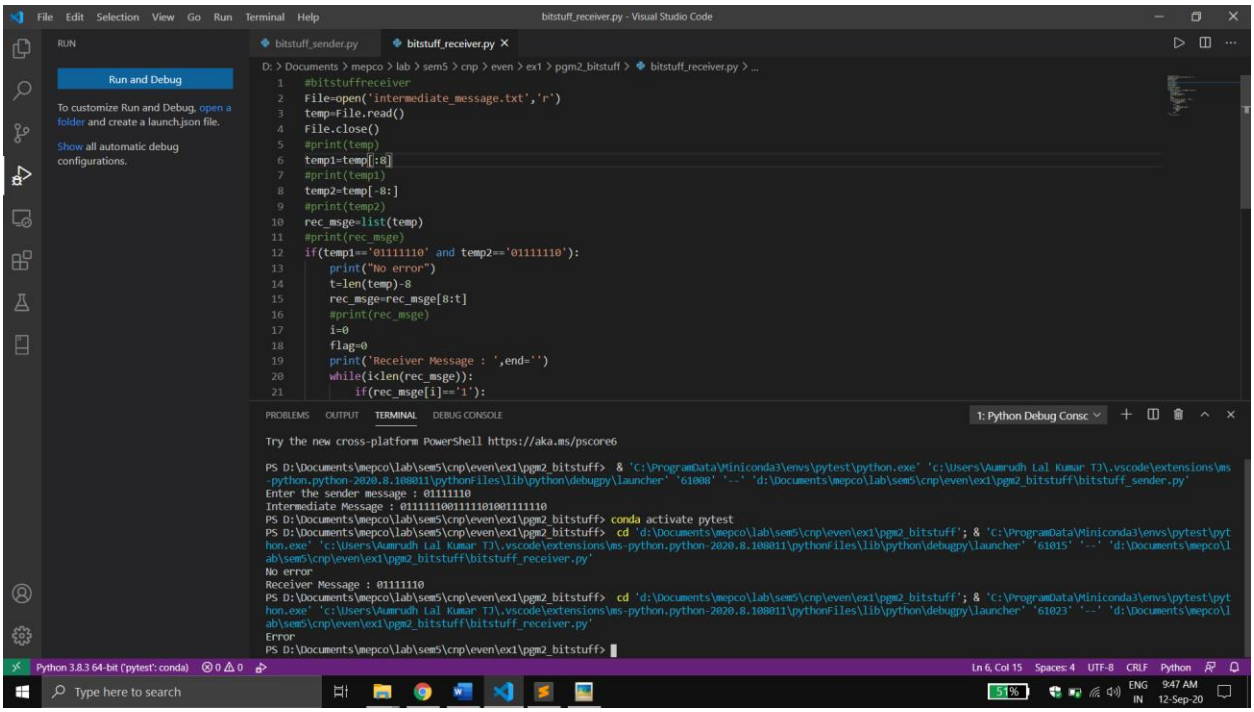
Receiver



With error: last 0 is removed in text file



Receiver (error)



Result:

The bit stuffing done in data link layer using framing was successfully programmed, tested and worked fine in python. The intermediate message was stored with 0111110 stuffed before and after message. Further if the sender message had continuous 5 one's then a zero was added after 5 1's. The Receiver received the original message sent by sender.