**The Whole Code:**

GTask\_Two = open('LastVersion.txt','r',encoding='utf-8').read()

#Firstly read the file what we already have from the previous step in output

Gdata\_bits = 4

#Next,we set our data bit equal to 4 due to\_(7,4)

GFunc = [GTask\_Two[i:i+Gdata\_bits] for i in range(0, len(GTask\_Two), Gdata\_bits)]

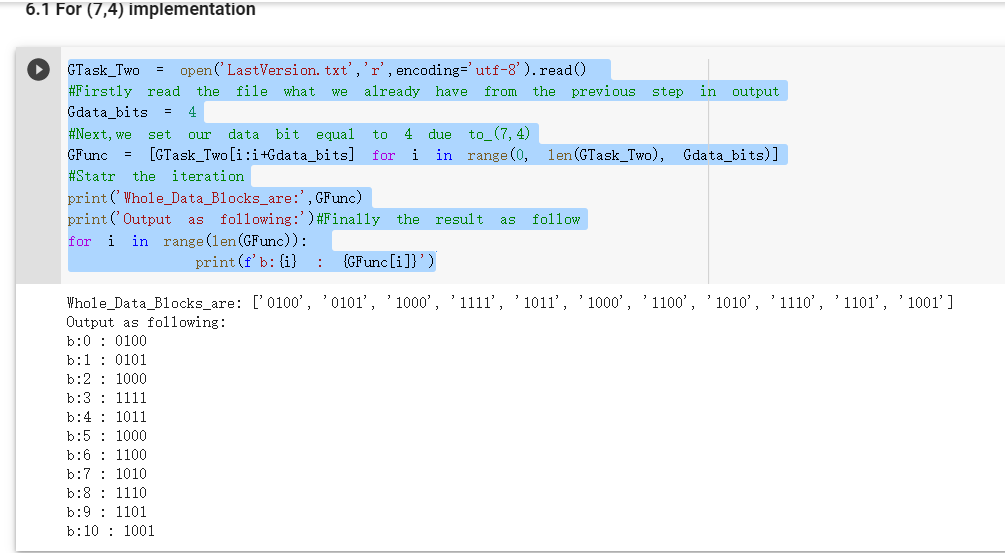
#Statr the iteration

print('Whole\_Data\_Blocks\_are:',GFunc)

print('Output as following:')#Finally the result as follow

for i in range(len(GFunc)):

        print(f'b:{i} : {GFunc[i]}')



GTask\_Two = open('LastVersion.txt','r',encoding='utf-8').read()

#Firstly read the file what we already have from the previous step in output

Gdata\_bits = 11

#Next,we set our data bit equal to 11 due to\_(11,15)

GFunc2 = [GTask\_Two[i:i+Gdata\_bits] for i in range(0, len(GTask\_Two), Gdata\_bits)]

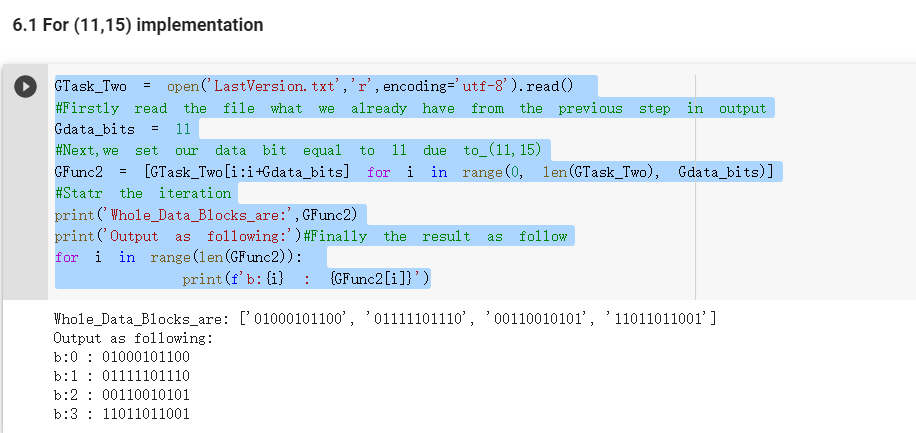
#Statr the iteration

print('Whole\_Data\_Blocks\_are:',GFunc2)

print('Output as following:')#Finally the result as follow

for i in range(len(GFunc2)):

        print(f'b:{i} : {GFunc2[i]}')



def Gul\_HammingEncode\_74\_Imp(ttheBitString):

#Then defined our function related to 74 here

    GTotalNumber = 1

    GWholeeBitNum = [ttheBitString[i:i+GTotalNumber] for i in range(0, len(ttheBitString), GTotalNumber)]

    #Iteration step

    for i in range(0, len(GWholeeBitNum)):

      #loop

      GWholeeBitNum[i] = int(GWholeeBitNum[i])

    #Get the result

    print(f'BitString\_{ttheBitString}:')

    #As the assignment given

    print(f'Expand the block to 8 bits: \_ \_ \_ {GWholeeBitNum[0]} \_ {GWholeeBitNum[1]} {GWholeeBitNum[2]} {GWholeeBitNum[3]}.')

    p1 = GWholeeBitNum[0]^GWholeeBitNum[1]^GWholeeBitNum[3]

    print(f'p1: b3 + b5 + b7 = {GWholeeBitNum[0]}+{GWholeeBitNum[1]}+{GWholeeBitNum[3]} = {p1}.')

    p2 = GWholeeBitNum[0]^GWholeeBitNum[2]^GWholeeBitNum[3]

    print(f'p2: b3 + b6 + b7 = {GWholeeBitNum[0]}+{GWholeeBitNum[2]}+{GWholeeBitNum[3]} = {p2}.')

    p3 = GWholeeBitNum[1]^GWholeeBitNum[2]^GWholeeBitNum[3]

    print(f'p3: b5 + b6 + b7 = {GWholeeBitNum[1]}+{GWholeeBitNum[2]}+{GWholeeBitNum[3]} = {p3}.')

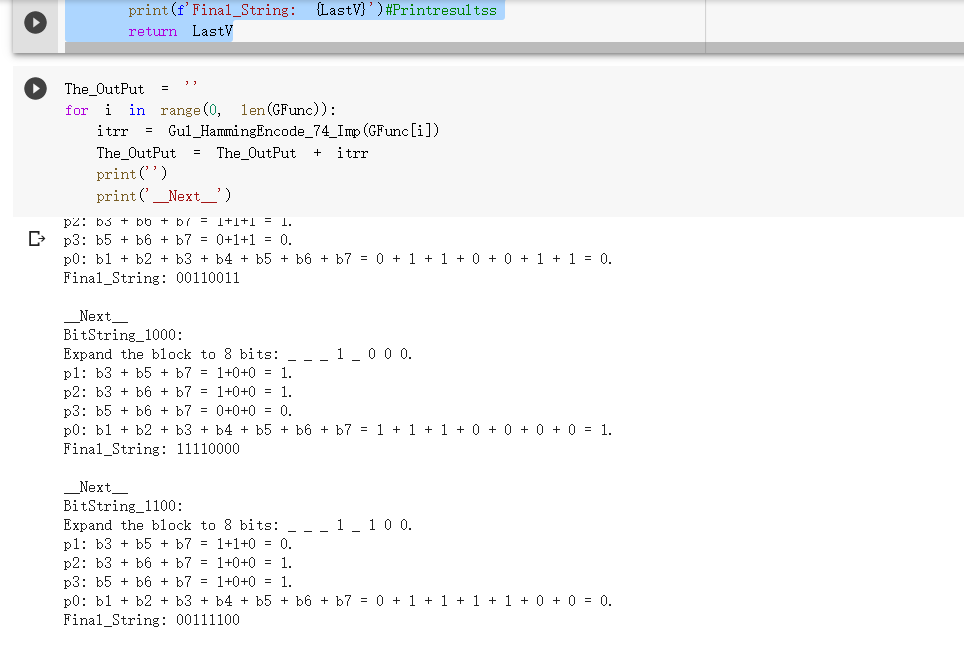
    p0 = p1^p2^GWholeeBitNum[0]^p3^GWholeeBitNum[1]^GWholeeBitNum[2]^GWholeeBitNum[3]

    print(f'p0: b1 + b2 + b3 + b4 + b5 + b6 + b7 = {p1} + {p2} + {GWholeeBitNum[0]} + {p3} + {GWholeeBitNum[1]} + {GWholeeBitNum[2]} + {GWholeeBitNum[3]} = {p0}.')

    LastV = "%d%d%d%d%d%d%d%d" % (p0,p1,p2,GWholeeBitNum[0],p3,GWholeeBitNum[1],GWholeeBitNum[2],GWholeeBitNum[3])

    print(f'Final\_String: {LastV}')#Printresultss

return LastV



def Gul\_HammingEncode\_1115\_Imp(ttheBitString2):

#Then defined our function related to 74 here

    GTotalNumber = 1

    GWholeeBitNum = [ttheBitString2[i:i+GTotalNumber] for i in range(0, len(ttheBitString2), GTotalNumber)]

    #Iteration step

    for i in range(0, len(GWholeeBitNum)):

      #loop

      GWholeeBitNum[i] = int(GWholeeBitNum[i])

    #Get the result

    print(f'BitString\_{ttheBitString2}:')

    #As the assignment given

    print(f'Expand the block to 16 bits: \_ \_ \_ {GWholeeBitNum[0]} \_ {GWholeeBitNum[1]} {GWholeeBitNum[2]} {GWholeeBitNum[3]} \_ {GWholeeBitNum[4]} {GWholeeBitNum[5]} {GWholeeBitNum[6]}{GWholeeBitNum[7]}{GWholeeBitNum[8]}{GWholeeBitNum[9]} {GWholeeBitNum[10]} .')

    p1 = GWholeeBitNum[0]^GWholeeBitNum[1]^GWholeeBitNum[3]^GWholeeBitNum[4]^GWholeeBitNum[6]^GWholeeBitNum[8]^GWholeeBitNum[10]

    print(f'p1: b3 + b5 + b7+ b9 + b11 + b13 + b15  = {GWholeeBitNum[0]}+{GWholeeBitNum[1]}+{GWholeeBitNum[3]}+{GWholeeBitNum[4]}+{GWholeeBitNum[6]}+{GWholeeBitNum[8]}+{GWholeeBitNum[10]} = {p1}.')

    p2 = GWholeeBitNum[0]^GWholeeBitNum[2]^GWholeeBitNum[3]^GWholeeBitNum[5]^GWholeeBitNum[6]^GWholeeBitNum[9]^GWholeeBitNum[10]

    print(f'p2: b3 + b6 + b7 + b10 + b11 + b14 + b15 = {GWholeeBitNum[0]}+{GWholeeBitNum[2]}+{GWholeeBitNum[3]}+{GWholeeBitNum[5]}+{GWholeeBitNum[6]}+{GWholeeBitNum[9]}+{GWholeeBitNum[10]} = {p2}.')

    p3 = GWholeeBitNum[1]^GWholeeBitNum[2]^GWholeeBitNum[3]^GWholeeBitNum[7]^GWholeeBitNum[8]^GWholeeBitNum[9]^GWholeeBitNum[10]

    print(f'p3: b5 + b6 + b7 + b12 + b13 + b14 + b15 = {GWholeeBitNum[1]}+{GWholeeBitNum[2]}+{GWholeeBitNum[3]}+{GWholeeBitNum[7]}+{GWholeeBitNum[8]}+{GWholeeBitNum[9]}+{GWholeeBitNum[10]} = {p3}.')

    p4 = GWholeeBitNum[4]^GWholeeBitNum[5]^GWholeeBitNum[6]^GWholeeBitNum[7]^GWholeeBitNum[8]^GWholeeBitNum[9]^GWholeeBitNum[10]

    print(f'p4: b9 + b10 + b11 + b12 + b13 + b14 + b15 = {GWholeeBitNum[4]}+{GWholeeBitNum[5]}+{GWholeeBitNum[6]}+{GWholeeBitNum[7]}+{GWholeeBitNum[8]}+{GWholeeBitNum[9]}+{GWholeeBitNum[10]} = {p3}.')

    p0 = p1^p2^GWholeeBitNum[0]^p3^GWholeeBitNum[1]^GWholeeBitNum[2]^GWholeeBitNum[3]^p4^GWholeeBitNum[4]^GWholeeBitNum[5]^GWholeeBitNum[6]^GWholeeBitNum[7]^GWholeeBitNum[8]^GWholeeBitNum[9]^GWholeeBitNum[10]

    print(f'p0: b1 + b2 + b3 + b4 + b5 + b6 + b7 + b8 + b9 + b10 + b11 + b12 + b13 + b14 + b15 = {p1} + {p2} + {GWholeeBitNum[0]} + {p3} + {GWholeeBitNum[1]} + {GWholeeBitNum[2]} + {GWholeeBitNum[3]} + {p4} + {GWholeeBitNum[4]} + {GWholeeBitNum[5]} +  {GWholeeBitNum[6]} + {GWholeeBitNum[7]} +  {GWholeeBitNum[8]} + {GWholeeBitNum[9]} +  {GWholeeBitNum[10]}  = {p0}.')

    LastV2 = "%d%d%d%d%d%d%d%d%d%d%d%d%d%d%d%d" % (p0,p1,p2,GWholeeBitNum[0],p3,GWholeeBitNum[1],GWholeeBitNum[2],GWholeeBitNum[3],p4,GWholeeBitNum[4],GWholeeBitNum[5],GWholeeBitNum[6],GWholeeBitNum[7],GWholeeBitNum[8],GWholeeBitNum[9],GWholeeBitNum[10])

    print(f'Final\_String: {LastV2}')#Printresultss

    return LastV2

