ASSIGNMENT 6

Q1.) **CODE:**

```
def aacomp(myseq):
   sequence=list(myseq)
   aa1 = list("ACDEFGHIKLMNPQRSTVWY")
   index=[]
   A=[None]*20
   for i in aa1:
      index.append(sequence.count(i))
   for i in range(20):
      A[i]= (aa1[i],(index[i]/len(myseq))*100)
   return(A)
print(aacomp('QSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAGYTYWEGGDEGAGKHSLSFAPVFVYEFAGDSIKPFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRIGAGLKFANGQSVGVRAIHYSNAGLKQPNDGIESYSLFYKIPI'))
```

OUTPUT:

Sequence 1:

```
[('A', 17.91044776119403), ('C', 1.4925373134328357), ('D', 2.9850746268656714), ('E', 2.9850746268656714), ('F', 1.4925373134328357), ('G', 5.970149253731343), ('H', 4.477611940298507), ('I', 1.4925373134328357), ('K', 7.462686567164178), ('L', 8.955223880597014), ('M', 0.0), ('N', 1.4925373134328357), ('P', 7.462686567164178), ('Q', 1.4925373134328357), ('R', 5.970149253731343), ('S', 5.970149253731343), ('T', 10.44776119402985), ('V', 1.4925373134328357), ('W', 5.970149253731343), ('Y', 4.477611940298507)]
```

Sequence 2:

```
[('A', 19.11764705882353), ('C', 0.0), ('D', 2.941176470588235), ('E', 1.4705882352941175), ('F', 5.88235294117647), ('G', 14.705882352941178), ('H', 0.0), ('I', 11.76470588235294), ('K', 1.4705882352941175), ('L', 13.23529411764706), ('M', 7.352941176470589), ('N', 0.0), ('P', 4.411764705882353), ('Q', 2.941176470588235), ('R', 2.941176470588235), ('S', 0.0), ('T', 1.4705882352941175), ('V', 8.823529411764707), ('W', 0.0), ('Y', 1.4705882352941175)]
```

Sequence 3:

```
[('A', 7.913669064748201), ('C', 0.0), ('D', 5.755395683453238), ('E', 4.316546762589928), ('F', 5.755395683453238), ('G', 15.107913669064748), ('H', 1.4388489208633095), ('I', 5.755395683453238), ('K', 4.316546762589928), ('L', 6.474820143884892), ('M', 0.7194244604316548), ('N', 3.597122302158273), ('P', 2.877697841726619), ('Q', 3.597122302158273), ('R', 3.597122302158273), ('S', 10.071942446043165), ('T', 4.316546762589928), ('V', 4.316546762589928), ('W', 4.316546762589928), ('Y', 5.755395683453238)]
```

Q2.) **CODE:**

```
def molwt(myseq):
   wt= {'A': 85 , 'C': 115, 'D': 130,'E': 145, 'F': 160 ,'G': 70, 'W': 200,
        'H': 150,'I': 125, 'K': 145, 'L': 125, 'M': 143, 'N': 130, 'Y': 175,
        'P': 110 ,'Q': 140, 'R': 170, 'S': 100, 'T': 115 ,'V': 110}
   length = len(myseq)
   mw = 0
   for i in range(0,length):
      mw= mw + wt[myseq[i]]
   mw= mw - (length -1)*18
   return(mw)
print(molwt('QSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAGYTYWEGGDEGAGKHSLSFAPVFVYEFAGDSIKPFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRIGAGLKFANGQSVGVRAIHYSNAGLKQPNDGIESYSLFYKIPI'))
```

OUTPUT:

Sequence 1: 7127

Sequence 2: 6529

Sequence 3: 14539

Q3.) **CODE:**

```
def comppro(myseq):
   sequence=list(myseq)
   aa1 = list("ADCETFGHIVKLMNWPQRSY")
   index=[]
   pro=[None]*20

   for i in aa1:
      index.append(sequence.count(i))
   for i in range(20):
      pro[i]= (index[i]/len(myseq)*100)

   compA= [8.47,5.97,1.39,6.32,5.79,3.91,7.82,2.26, 5.71,7.02,
       5.76,8.48, 2.21, 4.54,1.44,4.63,3.82,4.93,5.94,3.58]
   diffA = [compA - pro for compA,pro in zip(compA,pro)]
   diffA = [abs(ele) for ele in diffA ]
   sigmaA= sum(diffA)

   compB= [ 8.95,5.91,0.47,4.78,6.54,3.68,8.54,1.25,4.77,6.76,
       4.93, 8.78,1.56,5.74,1.24,3.74,4.75,5.24,8.05,4.13]
   diffB = [compB- pro for compB,pro in zip(compB,pro)]
   diffB = [abs(ele) for ele in diffB ]
   sigmaB= sum(diffB)

   if sigmaA> sigmaB :
     return('the sequence belongs to group B')
   else:
     return('the sequence belongs to group A')
print(comppro('RATPTRWPVGCFNRPWTKWSYDEALDGIKAAGYAWTGLLTASKPSLHHATATPEYLAALKQKSRHAA'))
```

OUTPUT:

Sequence 1: the sequence belongs to group A

Sequence 2: the sequence belongs to group A

Sequence 3: the sequence belongs to group B

Q4.)

CODE:

```
import numpy as np
def pref(seq):
    pair1 = np.zeros([20,20])
    pair2 = np.zeros([20,20])
    pair3 = np.zeros([20,20])
    aadict = {"A":0,"G":1,"V":2,"I":3,"L":4,"F":5,"Y":6,"W":7,"D":8,"E":9,"R":10,"N":11,"K":12,"S":13,"T":14,"C":15,"M":16,"H":17,"Q":18,"P":19}
    length = len(seq)


    for i in range(length-1):
        pair1[aadict[seq[i]]][aadict[seq[i+1]]] += 100/(seq.count(seq[i])+seq.count(seq[i+1]))
        pair2[aadict[seq[i]]][aadict[seq[i+1]]] += 100/(length-1)
        pair3[aadict[seq[i]]][aadict[seq[i+1]]] += 100/(seq.count(seq[i])*seq.count(seq[i+1]))


    return pair1,pair2,pair3
print(pref('RATPTRWPVGCFNRPWTKWSYDEALDGIKAAGYAWTGLLTASKPSLHHATATPEYLAALKQKSRHAA'))
```

OUTPUT listing the top 10 preferred residues

Sequence no	Pair 1	Pair2	Pair3
1	CF,FN,DE,EY,NR,GC ,VG,YD,GI,WT	AA,AT,AL,TA,TP,WT, HA,CF,LL,LD	CF,FN,NR,VG,DE, GI,GC,KQ,IK,QK
2	TQ,RT,VM,RQ,GL, PD,KF,QP,AA,IP	AA,GL,AV,AI,IG, VM,MG,GI,LG,GA	RT,TQ,RQ,KF,QP, PD,YV,QF,LY,LE
3	WD,AG,FA,YW,RL MT,TY,IE,LK,SL	AG,WD,SL,GL,GD,FA, VG,LG,YS,LK	MT,KH,HY,IH,WD, PN,QP,GM,RL,TY

Q5.)

CODE FOR HYDROPHOBICITY:

```
def Hgm(myseq):
    hgmlist = {'A':13.85,'D':11.61,'C':15.37,'E':11.38,'F':13.93,'G':13.34,
          'H':13.82,'I':15.28,'K':11.58,'L':14.13,'M':13.86,'N':13.02,'P':12.35,'Q':12.61,
          'R':13.10,'S':13.39,'T': 12.70,'V':14.56,'W':15.48,'Y':13.88}
    length = len(myseq)
    Hgm = 0
    for i in range(0,length):
        Hgm= Hgm + hgmlist[myseq[i]]
    return(Hgm/length)
print(Hgm('QSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAGYTYWEGGDEGAGKHSLSFAPVFVYEFAGDSIKPFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRIGAGLKFANGQSVGVRAIHYSNAGLKQPNDGIESYSLFYKIPI'))
```

CODE FOR HELICAL CONTACT AREA:

```
def ca(myseq):
    calist = {'A':20,'D':26,'C':25,'E':33,'F':46,'G':13,
          'H':37,'I':39,'K':46,'L':35,'M':43,'N':28,'P':22,'Q':36,
          'R':55,'S':20,'T': 28,'V':33,'W':61,'Y':46}
    length = len(myseq)
    ca = 0
    for i in range(0,length):
        ca= ca + calist[myseq[i]]
    return(ca)
print(ca('QSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAGYTYWEGGDEGAGKHSLSFAPVFVYEFAGDSIK
PFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRIGAGLKFANGQSVGVRAIHYSNAGLKQPNDGIESYSLFYKIPI'))
```

CODE FOR TOTAL NON BONDED ENERGY:

```
def et(myseq):
    etlist = {'A':1.90,'D':1.52,'C':2.04,'E':1.54,'F':1.86,'G':1.90,
          'H':1.76,'I':1.95,'K':1.37,'L':1.97,'M':1.96,'N':1.56,'P':1.70,'Q':1.52,
          'R':1.48,'S':1.75,'T':1.77,'V':1.98,'W':1.87,'Y':1.69}
    length = len(myseq)
    et = 0
    for i in range(0,length):
        et = et + etlist[myseq[i]]
    return(et)
print(et('AAAVMMGLAAIGAAIGIGILGGKFLEGAARQPDLIPLLRTQFFIVMGLVDAIPMIAVGLGLYVMFAVA'))
```

OUTPUT SUMMARISED:

properties	Seq1	Seq2	Seq3
hydrophobicity	13.35	13.77	13.41
Helical contact area	2156	2067	4350
Total non bonded energy	117.74000000000005	126.66000000000003	245.45000000000016