BIOINFORMATICS
PRACTICAL 7

KUDARI PAVANI
BE19B023

1.

```
def Que_1(seq):
    A_a = {'A':0, 'C':0, 'D':0, 'E':0, 'F':0, 'G':0, 'H':0, 'I':0, 'K':0,'L':0,'M':0, 'N':0, 'P':0, 'Q':0, 'R':0, 'S':0, 'T':0, 'V':0, 'W':0,'Y':0}
    for i in seq:
        if i in A_a:
            A_a[i] += 1*(100/len(seq))
    return A_a
if __name__ == "__main__":
    seq1 ='RATPTRWPVGCFNRPWTKWSYDEALDGIKAAGYAWTGLLTASKPSLHHATATPEYLAALKQKSRHAA'
    seq1 = Que_1(seq1)
    seq2 ='AAAVMMGLAAIGAAIGIGILGGKFLEGAARQPDLIPLLRTQFFIVMGLVDAIPMIAVGLGLYVMFAVA'
    seq2 = Que_1(seq2)
    seq3 ='AADVSAAVGATGQSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAGYTYWEGGDEGAGKHSLSFAPVFVYEFAGDSIKPFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRIGAGLKFANGQSVGVRAIHYSNAGLKQPNDGIESYSLFYKIPI'
    seq3 = Que_1(seq3)
    st = "{:<10} {:<20} {:<20} {:<20}"
    print(st.format('Amino acids', 'Sequence 1', 'Sequence 2', 'Sequence3'))
    A = ['A', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L','M', 'N', 'P', 'Q', 'R', 'S', 'T', 'V', 'W', 'Y']
    for j in range(len(A)):
        print(st.format(A[j], seq1[A[j]], seq2[A[j]], seq3[A[j]]))
```

Amino acids	Sequence 1	Sequence 2	Sequence3
A	17.910447761194032	19.11764705882353	10.596026490066226
C	1.492537313432836	0	0
D	2.985074626865672	2.9411764705882355	5.960264900662251
E	2.985074626865672	1.4705882352941178	3.9735099337748343
F	1.492537313432836	5.882352941176471	5.298013245033112
G	5.970149253731344	14.705882352941178	15.231788079470205
H	4.477611940298508	0	1.3245033112582782
I	1.492537313432836	11.764705882352942	5.298013245033112
K	7.462686567164179	1.4705882352941178	3.9735099337748343
L	8.955223880597016	13.23529411764706	5.960264900662251
M	0	7.352941176470589	0.6622516556291391
N	1.492537313432836	0	3.3112582781456954
P	7.462686567164179	4.411764705882353	2.6490066225165565
Q	1.492537313432836	2.9411764705882355	3.3112582781456954
R	5.970149253731344	2.9411764705882355	3.3112582781456954
S	5.970149253731344	0	9.933774834437086
T	10.447761194029852	1.4705882352941178	4.635761589403973
V	1.492537313432836	8.823529411764707	5.298013245033112
W	5.970149253731344	0	3.9735099337748343
Y	4.477611940298508	1.4705882352941178	5.298013245033112

Observations:

- **Sequence 1:**
    - Alanine, Threonine, Leucine, Lysine, and Proline are all found in higher concentrations. Because of the presence of Threonine, the sequence is both hydrophobic and hydrophilic. In a typical hydrophilic environment, such as outer space, where water is abundant, this polypeptide sequence must be found in the inner core of a protein.
- **Sequence 2:**
    - Sequence 2 has a larger amount of Alanine, Glycine, Isoleucine, and Leucine, making it extremely hydrophobic. In a typical hydrophilic environment, such as outer space, where water is abundant, this polypeptide sequence must be found in the inner core of a protein.
- **Sequence 3:**
    - Sequence 3 has higher levels of Glycine, Serine, and Alanine. Despite the inclusion of Gly and Ala, which would normally make this protein hydrophobic, serine gives it a modest hydrophilicity.

2.

```
def Que_2(seq):
    A_a_m_w = {'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}
    m_w = 0
    for i in seq:
        m_w += A_a_m_w[i]
    m_w -= (len(seq) - 1) * 18
    return m_w
if __name__ == "__main__":
    seq1 ='RATPTRWPVGCFNRPWTKWSYDEALDGIKAAGYAWTGLLTASKPSLHHATATPEYLAALKQKSRHAA'
    seq2 ='AAAVMMGLAAIGAAIGIGILGGKFLEGAARQPDLIPLLRTQFFIVMGLVDAIPMIAVGLGLYVMFAVA'
    seq3 ='AADVSAAVGATGQSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAGYTYWEGGDEGAGKHSLSFAPVFVYEFAGDSIKPFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRIGAGLKFANGQSVGVRAIHYSNAGLKQPNDGIESYSLFYKIPI'
    i = 0
    for k in [seq1, seq2, seq3]:
        i += 1
        print(f"The molecular weight for seq{i} = {Que_2(k)} units.")
```

```
The molecular weight for seq1 = 7127 units.
The molecular weight for seq2 = 6529 units.
The molecular weight for seq3 = 15453 units.
```

3.

```
def Que_1(seq):
    A_a = {'A':0, 'C':0, 'D':0, 'E':0, 'F':0, 'G':0, 'H':0, 'I':0, 'K':0,'L':0,'M':0, 'N':0, 'P':0, 'Q':0, 'R':0, 'S':0, 'T':0, 'V':0, 'W':0,'Y':0}
    for i in seq:
        if i in A_a:
            A_a[i] += 1*(100/len(seq))
    return A_a

def Que_3(seq):
    Group_A = {'A': 8.47, 'D': 5.97, 'C': 1.39, 'E': 6.32, 'T': 5.79,
               'F': 3.91, 'G': 7.82, 'H': 2.26, 'I': 5.71, 'V': 7.02,
               'K': 5.76, 'L': 8.48, 'M': 2.21, 'N': 4.54, 'W': 1.44,
               'P': 4.63, 'Q': 3.82, 'R': 4.93, 'S': 5.94, 'Y': 3.58}

    Group_B = {'A': 8.95, 'D': 5.91, 'C': 0.47, 'E': 4.78, 'T': 6.54,
               'F': 3.68, 'G': 8.54, 'H': 1.25, 'I': 4.77, 'V': 6.76,
               'K': 4.93, 'L': 8.78, 'M': 1.56, 'N': 5.74, 'W': 1.24,
               'P': 3.74, 'Q': 4.75, 'R': 5.24, 'S': 8.05, 'Y': 4.13}

    grp_A, grp_B = 0, 0
    comp = Que_1(seq)
    A_a = ['A', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L','M', 'N', 'P', 'Q', 'R', 'S', 'T', 'V', 'W', 'Y']
    for i in A_a:
        grp_A += abs(comp[i] - Group_A[i])
        grp_B += abs(comp[i] - Group_B[i])
    return grp_A, grp_B

if __name__ == "__main__":
    seq1 = 'RATPTRWPVGCFNRPWTKWSYDEALDGIKAAGYAWTGLLTASKPSLHHATATPEYLAALKQKSRHAA'
    seq2 = 'AAAVMMGLAAIGAAIGIGILGGKFLEGAARQPDLIPLLRTQFFIVMGLVDAIPMIAVGLGLYVMFAVA'
    seq3 = 'AADVSAAVGATGQSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAGYTYWEGGDEGAGKHSLSFAPVFVYEFAGDSIKPFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRIGAGLKFANGQSVGVRAIHYSNAGLKQPNDGIESYSLFYKIPI'

    for k in [seq1, seq2, seq3]:
        dev_A, dev_B = Q3(k)
        print(f"Deviation score from Group A: {dev_A}.\n" f"Deviation score from Group B: {dev_B}.")
        if dev_A > dev_B:
            print(f"\nThe given sequence is closer to Group B.\n")
        else:
            print(f"\nThe given sequence is closer to Group A.\n")
```

Output:

Deviation score from Group A: 55.84402985074628.
Deviation score from Group B: 58.523731343283586.

The given sequence is closer to Group A.

Deviation score from Group A: 74.51470588235291.
Deviation score from Group B: 76.83823529411764.

The given sequence is closer to Group A.

Deviation score from Group A: 38.33225165562914.
Deviation score from Group B: 32.597814569536425.

The given sequence is closer to Group B.

4)The top preferred pairs for the given sequence, for each formula are:

Formula-1:
['WD', 'IE', 'RL', 'YW', 'SL', 'QP', 'QS', 'TG', 'VG', 'WQ']

Formula-2:
['AG', 'FA', 'GA', 'GD', 'GL', 'SL', 'TG', 'VG', 'WD', 'IG']

Formula-3:
['MT', 'IH', 'WD', 'QP', 'RL', 'YW', 'WQ', 'QT', 'TR', 'WE']

5)

- The average hydrophobicity is calculated by multiplying the "Hgm" values for each amino acid from the database by the total number of amino acids.
- The helical contact area is calculated by summing the "Ca" values for each amino acid from the database.
- The total non-bonded energy is derived by adding the "Et" values for each amino acid from the database.

```python
def Que_5(seq):
    Hgm = {'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}
    Ca = {'A': 20.00, 'D': 26.00, 'C': 25.00, 'E': 33.00, 'F': 46.00,
          'G': 13.00, 'H': 37.00, 'I': 39.00, 'K': 46.00, 'L': 35.00,
          'M': 43.00, 'N': 28.00, 'P': 22.00, 'Q': 36.00, 'R': 55.00,
          'S': 20.00, 'T': 28.00, 'V': 33.00, 'W': 61.00, 'Y': 46.00}
    Et = {'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}
    avg_Hgm = 0
    ca = 0
    et = 0
    for i in range(len(seq)):
        avg_Hgm += (Hgm[seq[i]] / len(seq))
        ca += Ca[seq[i]]
        et += Et[seq[i]]
    print(f"Average hydrophobicity = {avg_Hgm}"
          f"\nHelical contact area = {ca}"
          f"\nTotal non-bonded energy = {et}\n")
if __name__ == "__main__":
    seq1 ='RATPTRWPVGCFNRPWTKWSYDEALDGIKAAGYAWTGLLTASKPSLHHATATPEYLAALKQKSRHAA'
    seq2 ='AAAVMMGLAAIGAAIGIGILGGKFLEGAARQPDLIPLLRTQFFIVMGLVDAIPMIAVGLGLYVMFAVA'
    seq3 ='AADVSAAVGATGQSGMTYRLGLSWDWDKSNWQTSTGRLTGYWDAGYTYWEGGDEGAGKHSLSFAPVFVYEFAGDSIKPFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRIGAGLKFANGQSVGVRAIHYSNAGLKQPNDGIESYSLFYKIPI'
    for j in [seq1, seq2, seq3]:
        Que_5(j)
```

```
Average hydrophobicity for seq[i] = 13.352537313432833
Helical contact area = 2156.0
Total non-bonded energy = 117.74000000000005

Average hydrophobicity for seq[i] = 13.77161764705882
Helical contact area = 2067.0
Total non-bonded energy = 126.66000000000003

Average hydrophobicity for seq[i] = 13.418675496688742
Helical contact area = 4616.0
Total non-bonded energy = 267.7500000000001
```

- The hydrophobicity of all of the aforementioned sequences is similar. This is because all polypeptide sequences have a large proportion of Alanine and other aliphatic amino acids.
- The helix contact area of sequence 3 is significantly bigger than that of sequences 1 and 2. Because sequence 3 might have a right-handed alpha-helical structure, whereas other peptides could have a beta-sheet structure, this is the case.

- The total non-bonded energy of sequence 3 is higher than that of sequences 1 and 2. Substantial values of "Et" for Gly and Ser in sequence 3 result in a large amount of total non-bonded energy.