

## Problem set 2: Strings and lists

1. A string is simply an ordered collection of symbols selected from some alphabet and formed into a word; the length of a string is the number of symbols that it contains. Given a DNA sequence as a string, find frequency of 'A', 'C', 'G', and 'T'. The output should be exactly in the format as shown below.

**Input:**

ATTAGCA

**Output:**

A: 3, T: 2, G: 1, C: 1

2. RNA is also a string formed from the alphabet containing 'A', 'C', 'G', and 'U'. Given a DNA T, return the transcribed RNA.

**Input:**

GATGGAACCTTGACTACGTAAATT

**Output:**

GAUGGAACUUGACUACGUAAAUU

3. In DNA strings, symbols 'A' and 'T' are complements of each other, as are 'C' and 'G'. The reverse complement of a DNA string *s* is the string *sc* formed by reversing the symbols of *s*, then taking the complement of each symbol (e.g., the reverse complement of "GTCA" is "TGAC"). Find reverse complement for a given DNA sequence.

**Input:**

AAAACCCGGT

**Output:**

ACCGGGTTTT

4. The GC-content of a DNA string is given by the percentage of symbols in the string that are 'C' or 'G'. For example, the GC-content of "AGCTATAG" is 37.5%. For the given *n* DNA strings, return the ID of the sequence having highest GC content followed by the GC-content of that string.

**Input:**

3

CCTGCGGAAGATCGGCACTAGAATAGCCAGAACCGTTTCTCTGAGGCTTCCGCGCTTCCCTCCCACTAATAATTCTGAGG  
CCATCGGTAGCGCATCCTTAGTCCAATTAAGTCCCTATCCAGGCGCTCCGCGGAAGGTCTATATCCATTTGTCAGCAGACACGC  
CCACCCTCGTGGTATGGCTAGGCAATTCAGGAACCGGAGAACGCTTCAGACCAGCCCGGACTGGGAACCTGCGGGCAGTAGGTGGAAT

**Output:**

3, 60.919540

5. The standard weight assigned to each member of the 20-symbol amino acid alphabet is the monoisotopic mass of the corresponding amino acid. Given a protein string P, return the total weight of P.

A	71.03711	M	131.04049	G	57.02146	S	87.03203
C	103.00919	N	114.04293	H	137.05891	T	101.04768
D	115.02694	P	97.05276	I	113.08406	V	99.06841
E	129.04259	Q	128.05858	K	128.09496	W	186.07931
F	147.06841	R	156.10111	L	113.08406	Y	163.06333

**Input:**

SKADYEK

**Output:**

821.39

6. For DNA strings **s1** and **s2** having the same length, their transition/transversion ratio  $R(\mathbf{s1}, \mathbf{s2})$  is the ratio of the total number of transitions to the total number of transversions, where symbol substitutions are inferred from mismatched corresponding symbols. Given DNA strings **s1** and **s2** of equal length, return the transition/transversion ratio  $R(\mathbf{s1}, \mathbf{s2})$

Transition: Purine  $\rightarrow$  Purine and Pyrimidine  $\rightarrow$  Pyrimidine (A  $\leftrightarrow$  G and C  $\leftrightarrow$  T) conversion.

Transversion: Purine  $\rightarrow$  Pyrimidine and Pyrimidine  $\rightarrow$  Purine (A  $\leftrightarrow$  C or T and G  $\leftrightarrow$  C or T) conversion

**Input:**

```
GCAACGCACAACGAAAACCCCTTAGGGACTGGATTATTTTCGTGATCGTTGTAGTTATTGGAAGTACGGGCATCAACCCAGTT
TTATCTGACAAAGAAAGCCGTCACGCGCTGGATAATTTTCGCGATCGTGCTGGTTACTGGCGGTACGAGTGTTTCCTTTGGGT
```

**Output:**

1.21

7. Given a list consisting integers 0, 1, and 2, sort it without using any in-built functions.

**Input:**

0 2 1 2 0 1 0

**Output:**

0 0 0 1 1 2 2

8. Given two matrices of dimension  $n \times n$ , **A** and **B**, find their sum. Use list of list to store values of each matrix.

**Input:**

```
3
1 2 3      2 1 0
1 0 0      2 1 1
2 2 2      2 2 2
```

**Output:**

```
3 3 3
3 1 1
4 4 4
```

9. Given a string, print all possible permutations of the string. You can use python in-built features.

**Input:**

abc

**Output:**

abc acb bac bca cab cba

10. If **A** and **B** are sets, then their union **A $\cup$ B** is the set comprising every element of sets **A** and **B**; their intersection **A $\cap$ B** is the set of elements present in both **A** and **B**; and their set difference **A-B** is the set of elements in A but not in B. Furthermore, if **A** is a subset of another set **V**, then the set complement of A with respect to **V** is defined as the set **A $^c$ =V-A**. Given, A positive integer  $n$  and two subsets **A** and **B**, return six sets:

a. **A $\cup$ B**

b. **A $\cap$ B**

c. **A-B**

d. **B-A**

e. **A $^c$**

f. **B $^c$**

**Input:** (Sets A, B and V. Numbers are separated by whitespace.)

```
1 2 3 4 5
2 8 5 10
1 2 3 4 5 6 7 8 9 10
```

**Output:** (order and format of the output should be maintained)

```
{1, 2, 3, 4, 8, 5, 10}
{2, 5}
{1, 3, 4}
```

```
{8, 10}  
{6, 7, 8, 9, 10}  
{1, 3, 4, 6, 7, 9}
```