

Exercise

Programs should be written in the programming language Python according to the respective exercise description. The program must be correct in terms of syntax and semantic. If there exists a minimal solution using only a single pre-defined function, this function is not allowed. The tasks on this sheet are for additional exercise.

1. We define S_n for $n \in \mathbb{N}$ as:

$$S_n = \sum_{k=1}^n k^2 + \frac{k}{2} \quad (1)$$

- (a) Write a program that reads an integer number from standard input, assigns it to variable n and calculates S_n and prints the result.
 - (b) We assume, the user inputs a float. How do you check if $n \in \mathbb{N}$?
 - (c) In which run-time complexity class is your implementation of task a? Use the common rules to justify your answer.
 - (d) Define a function `sn` that performs the calculation given in equation 1. The function takes n as argument and returns the respective S_n as a float value to the caller. No recursive function allowed.
2. Write a recursive function `sn_rec` for the following equation. The function takes $n \in \mathbb{N}$ as argument and returns the respective S_n as a float value.

$$S(n) = \begin{cases} 1.5 & ; \text{ if } n = 1 \\ S(n-1) + n^2 + \frac{n}{2} & ; \text{ if } n > 1 \end{cases} \quad (2)$$

3. The file `nucleotides.txt` contains for ribonucleotide- and deoxyribonucleotide triphosphates (first column) the molecular weight (second column), absorbance maxima (second column) and the molar extinction coefficients (third column)¹.
 - (a) Write a program that reads the file and stores the data in a dictionary (key=nucleotide, value=tuple containing molecular weight, absorbance maximum, and molar extinction coefficient).
 - (b) Based on the fully initialized dictionary, write a program that prints key-value pairs of the dictionary sorted by nucleotide in reverse order.

Example:

```
dTTP (482.2, 267.0, 9600.0)
dGTP (507.2, 253.0, 13700.0)
dCTP (467.2, 271.0, 9300.0)
dATP (491.2, 259.0, 15200.0)
UTP (484.2, 262.0, 10000.0)
GTP (523.2, 253.0, 13700.0)
CTP (483.2, 271.0, 9000.0)
ATP (507.2, 259.0, 15400.0)
```

¹<https://www.thermofisher.com/de/de/home/references/ambion-tech-support/rna-tools-and-calculators/dna-and-rna-molecular-weights-and-conversions.html>

- (c) Based on the fully initialized dictionary, write a program that prints key-value pairs of the dictionary sorted by molecular weight.

Example:

```
dCTP (467.2, 271.0, 9300.0)
dTTP (482.2, 267.0, 9600.0)
CTP (483.2, 271.0, 9000.0)
UTP (484.2, 262.0, 10000.0)
dATP (491.2, 259.0, 15200.0)
ATP (507.2, 259.0, 15400.0)
dGTP (507.2, 253.0, 13700.0)
GTP (523.2, 253.0, 13700.0)
```

- (d) Create a DataFrame containing the data, where columns consist of molecular weight, absorbance maximum, and molar extinction coefficient, while rows are respective amino acids. Include row and column labels (see example).

	Molecular weight (g/mol)	Absorbance Maximum at pH 7.0 (nm)	Molar Extinction Coefficient at pH 7.0
Nucleotides			
ATP	507.2	259.0	15400.0
CTP	483.2	271.0	9000.0
GTP	523.2	253.0	13700.0
UTP	484.2	262.0	10000.0
dATP	491.2	259.0	15200.0
dCTP	467.2	271.0	9300.0
dGTP	507.2	253.0	13700.0
dTTP	482.2	267.0	9600.0

- (e) How could you transpose the DataFrame?

4. Let s be a ssDNA sequence. Write a program that calculates the molecular weight for sequence s based on the molecular weights of deoxyribonucleotide monophosphates present in table `dataframe.csv`. Take into account, that the molecular weight for each nucleotide must be decreased by 18 g/mol (H_2O) when part of a polynucleotide. You can load the DataFrame from the file `dataframe.csv` with the following command:

```
import pandas as pd
df = pd.read_csv("dataframe.csv.txt", sep="\t", index_col=0)
```

Total molecular weight should be rounded at one digit after decimal point.

Example:

```
s = "AGCTTTTCGTAC"
```

Output:

```
Molecular weight of ssDNA: 3673.4 g/mol
```

5. (a) Write a program that calculates the frequencies f^{pq} of all 2-mers of a given amino acid sequence s .

Example:

```
s = "MLYITMLWVM"
```

Output:

```
2-mers of s: {'ML': 2, 'LY': 1, 'YI': 1, 'IT': 1, 'TM': 1, 'LW': 1, 'WV': 1, 'VM': 1}
```

- (b) For any given 2-mer of sequence s , check if the respective 2-mer is also present in sequence t .

Example:

2-mers of s : {'ML': 2, 'LY': 1, 'YI': 1, 'IT': 1, 'TM': 1, 'LW': 1, 'WV': 1, 'VM': 1}

2-mers of t : {'GI': 1, 'IY': 1, 'YI': 1, 'IT': 1, 'TM': 1, 'ML': 1, 'LW': 1}

Output:

2-mer ML of s (2x) also present in t (1x)

2-mer YI of s (1x) also present in t (1x)

2-mer IT of s (1x) also present in t (1x)

2-mer TM of s (1x) also present in t (1x)

2-mer LW of s (1x) also present in t (1x)

6. You will find the program `debug.ipynb`. The programmer unfortunately introduced a error(s), so that the program is not correct. Locate the mistake and correct it.