1. There is a bacteria colony that grows exponentially. After each hour bacteria inoculum P_0 is added. r is the growth rate. The total colony is calculated by the following equation:

$$P_t = P_0 \sum_{i=1}^{t} r^i$$

a) Use standard input to ask for two float variables and one integer. Assign and store them as P_0, r and t.

b) Write a program that calculates P_t as a float and prints the result with 2 decimal places.

Example input: P_0 = 100, r = 1.1, t = 5

Example Output: P_t = 671.56

c) Describe the algorithm complexity of b) using the rules discussed in the lecture?

d) The bacteria inoculum is now not added every hour, but every h hours. Write a recursive program that calculates P_t as a float and prints the result with 2 decimal places.

Example input: P_0 = 100, r = 1.1, h = 2, t = 3

-> Hour 1: 100*1.1 = 110

Hour 2: no inoculum is added, 110*1.1 = 121

Hour 3: (121+100)*1.1 = 243.1

Example Output: P_t = 243.1

2. The file genetic_code.txt contains all amino acids and their conjugate nucleic acids.

a) Open the file and store its contents in a dictionary. The key should be a tuple of the bases and the corresponding value of the coded amino acid.

b) Sort and print the dictionary in order of the bases.

```
(A, A, A) K

(A, A, C) T

(A, A, G) K

…
```

c) Sort and print the dictionary in order of the amino acids.

(T, A, A) *

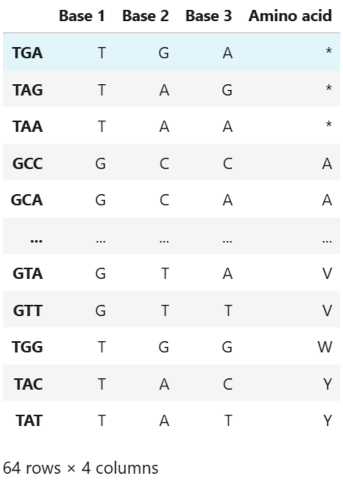
(T, A, G) *

(T, G, A) *

(G, C, A) A

…

d) Turn the dictionary into a dataframe. The first column should be the first base, the second one the second base, the third one the third base and the fourth the corresponding amino acid. Add an index column consisting of the concenated codon code per amino acid and proper column labels.



	Base 1	Base 2	Base 3	Amino acid
TGA	T	G	A	*
TAG	T	A	G	*
TAA	T	A	A	*
GCC	G	C	C	A
GCA	G	C	A	A
...	...	...	...	...
GTA	G	T	A	V
GTT	G	T	T	V
TGG	T	G	G	W
TAC	T	A	C	Y
TAT	T	A	T	Y

64 rows × 4 columns

3. file mit einer Tabelle

a) The data frame of 2) is stored in the file dataframe.csv.txt . Using the following code it can be imported. Using that, write a program that assigns amino acids to a string of nucleotides.

```
import pandas as pd

df = pd.read_csv(‘dataframe.csv.txt’, sep=’\t’, index_col=0)
```

(the ‘ won't work in jupyter so you have to type them again)

Example input: s = “ATGGTGGAACTA”

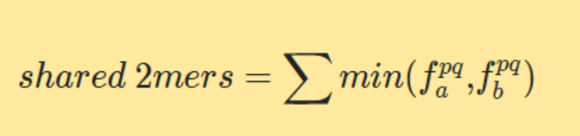
Example output: MVEL

(thats only near the one in the exam but the nucleotides match)

4. Lmers are all possible subsequences of a sequence consisting of l-chararacters. It is thought that the more lmers two sequences share the more identical they are. f^{pq} is the frequency of the lmers.

a) Write a program that counts all the different 2mers of sequence a and stores them in a dictionary. (example is in example of b for some reason)

b) Write a program that compares the 2mers of two sequences according to the following equation. f^{pq}_a and f^{pq}_b are the frequencies of the 2mers in the two sequences:



$$shared\ 2mers = \sum min(f_a^{pq}, f_b^{pq})$$

(not really the equation but it was smth with min and it was unnecessary because of the example)

Example input:

sequence a: MEIDJSEREINARFHUSEEREI

sequence b: MEIDFSEREINARFDUS (these are just random at this point)

Example output:

```
2mers of sequence a: {'ME': 1, 'EI': 3, 'ID': 1, 'DJ': 1, 'JS': 1, 'SE': 2, 'ER': 2, 'RE': 2, 'IN': 1, 'NA': 1, 'AR':
1, 'RF': 1, 'FH': 1, 'HU': 1, 'US': 1, 'EE': 1}

2mers of sequence b: {'ME': 1, 'EI': 2, 'ID': 1, 'DF': 1, 'FS': 1, 'SE': 1, 'ER': 1, 'RE': 1, 'IN': 1, 'NA': 1,
'AR': 1, 'RF': 1, 'FD': 1, 'DU': 1, 'US': 1}

shared 2mers: {'ME': 1, 'EI': 2, 'ID': 1, 'SE': 1, 'ER': 1, 'RE': 1, 'IN': 1, 'NA': 1, 'AR': 1, 'RF': 1, 'US': 1}

number of shared 2mers: 12
```

c) How many 2mers can two sequences of length l_a and l_b share at most?

5. debug a the file debug.ipynb