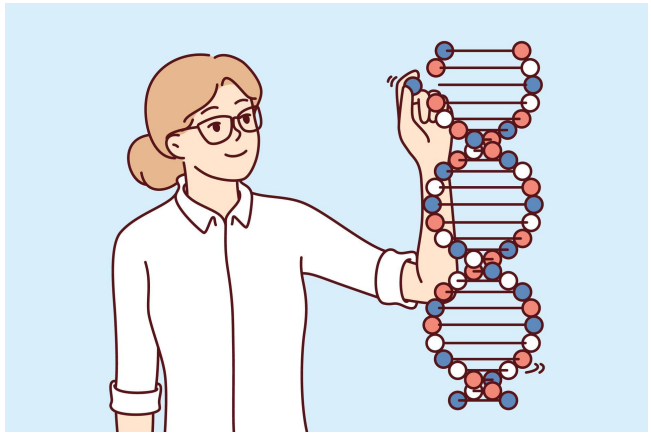
Week 6

Help the scientists code!

Python challenge

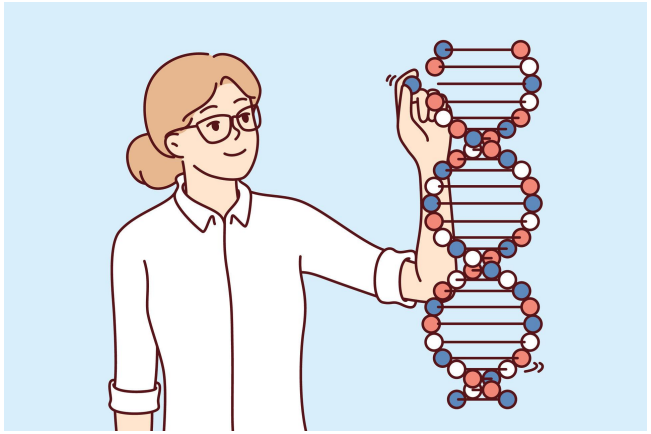


Gene sequences are made of the letters ‘A’, ‘T’, ‘C’, ‘G’

PROBLEM:

- Dr. Lily has a file called ‘seq_1.txt’.
- This file has gene sequences for four different organisms.
- She has noticed that the letters ‘TTA’ repeat throughout these gene sequences.
- She wants coding help to count how many times they repeat in each organism.

```
Human: ATGCGGGATTAGCATTAATGCCAATG
Mouse: AGCCTTATGAAATTACAGATGTTAAG
Bird:  TGACCATGGACTGATGTTAGGCAGGT
Dinosaur: CTGATTAATGTTAAAAGGCATGA
```



TASK 1:

- Read each line in seq_1.txt
- Print the sequence and the number of ‘TTA’s’ in that sequence

Hint: To count a certain letter or a pattern of letters in a string of letters, you can use something like this:

‘ATTAGC’.count(‘G’)

TASK 1:

- Read each line in seq_1.txt
- Print the sequence and the number of 'TTA's' in that sequence

Hint: To count a certain letter or a pattern of letters in a string of letters, you can use something like this:

'ATTAGC'.count('G')

Answer:

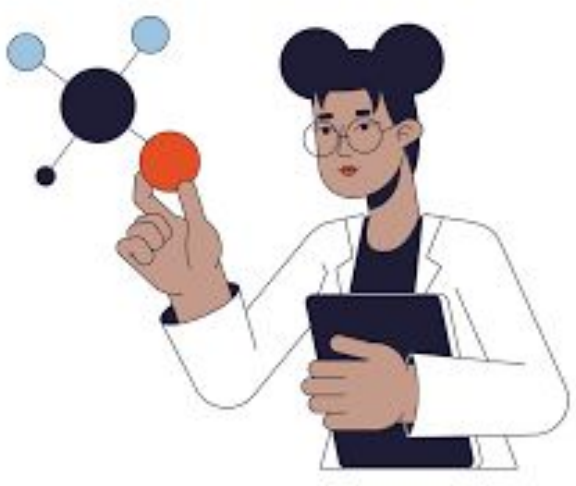
```
file_seq1 = open("seq_1.txt")
lines = file_seq1.readlines()

for seq in lines:
    TTA_count = seq.count('TTA')
    print(seq, TTA_count)
```

Genes go through a ‘decoding’ process so that they can make up proteins which support our body. This decoding process is called **‘transcription’**

Example: When a sequence ‘ATCG’ is transcribed, it becomes ‘UAGC’.

- The letter ‘A’ is always replaced by ‘U’
- The letter ‘T’ is always replaced by ‘A’
- The letter ‘C’ is always replaced by ‘G’
- The letter ‘G’ is always replaced by ‘C’

PROBLEM : Dr. Shauna has a file ‘seq_2.txt’ with longer gene sequences. She would like some coding help to transcribe them.



```
ATCGATCGATCGTACGTCAGTCGTACGTCAGTCAGA
ACTGACTGTACGTACGTACGATGCTACGTACGCATA
ACTACGTACGTACGTACGATCGTACGTACGCATACG
ATGCTACGTACGACGCTATCGATCGTACGTACGTAG
CTACGTGGGGGGAATATATTTGCGCGCCGTATAATA
TGCCGATATGCGGTGCCTCTCTCGGCGCGCGCATTT
TTGCGCGAAAAAAAAAGCGATCGATCGTACGAAAAA
TGCATAGCTAC
```

TASK 2:

- Read each line in the file ‘seq_2.txt’.
- Start with an empty string called ‘transcribed’. It will look like this:

```
transcribed = ' '
```

- To this line, add the corresponding letters for transcribing, as you read each letter in the line

For example:

```
For char in line:
    If char == 'A':
        Transcribed.append('U')
```

- Use a for loop and if/else statements
- For each line, print the transcribed line in the end

TASK 2:

- Read each line in the file ‘seq_2.txt’.
- Start with an empty string called ‘transcribed’. It will look like this:

```
transcribed = ' '
```

- To this line, add the corresponding letters for transcribing, as you read each letter in the line

For example:

```
For char in line:
    If char == 'A':
        Transcribed.append('U')
```

- Use a for loop and if/else statements
- For each line, print the transcribed line in the end

Answer:

```
file_seq2 = open("seq_2.txt")
lines = file_seq2.readlines()
for line in lines:
    transcribed = ' '
    for char in line:
        if char == 'A':
            transcribed = transcribed + 'U'
        elif char == 'T':
            transcribed = transcribed + 'A'
        elif char == 'C':
            transcribed = transcribed + 'G'
        elif char == 'G':
            transcribed = transcribed + 'C'
    print(transcribed)
```

TASK 3:

- Convert your code for transcription into a function.
- Use this function to transcribe ‘seq_3.txt’.

TASK 3:

- Convert your code for transcription into a function.
- Use this function to transcribe ‘seq_3.txt’.

ANSWER:

```
def transcribe_file(filename):
    file_seq = open(filename)
    lines = file_seq.readlines()
    for line in lines:
          transcribed = ''
          for char in line:
             if char == 'A':
             transcribed = transcribed + 'U'
             elif char == 'T':
                  transcribed = transcribed + 'A'
             elif char == 'C':
transcribed = transcribed + 'G'
                 elif char == 'G':
             transcribed = transcribed + 'C'
          print(transcribed)
```