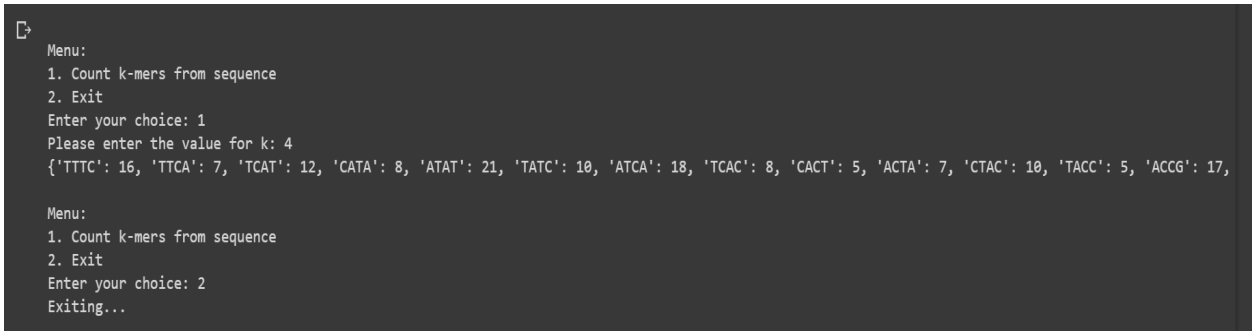
ABIN ASSIGNMENT GRP-6

1) The goal of this question is to count the occurrences of all k-mers in a given FASTA file containing nucleotide sequences.
Here is a detailed explanation of code too->

->Imports: Essential libraries for fetching sequence data from NCBI and processing it.
->Email & Accession Number: User's email is set for NCBI usage. A specific nucleotide sequence's accession number is hardcoded.
->download_sequence(): Retrieves the FASTA sequence from NCBI for the given accession number.
->count_kmers(): For a given sequence and k, it counts and returns all k-mers.
->count_kmers_from_string(): Processes FASTA formatted strings to count k-mers for each sequence and aggregates the results.
->main(): Orchestrates the overall process - downloads the sequence, gets user input for k, counts k-mers, and handles exceptions.
Execution Check: Ensures the script runs standalone and doesn't execute when imported.
Purpose: The script counts k-mers for an NCBI nucleotide sequence based on a user-specified k.



2) The goal in this question to find the edit distance between two sequences , edit distance is similar to hamming distance but unlike hamming distance we can perform insertion deletions and substitutions whereas in hamming distance we only see the number of changes or substitutions

Edit distance can be visualized and understood in a general sense by taking two strings as alpha , beta and x, y as the endings

Edit distance(source, target) = min(Edit distance(alpha,beta) Edit distance(alpha_x,beta) ,(alpha , beta_y))

Edit distance is calculated using a dynamic programming approach each cell in the matrix is filled by the up down and diagonal cell . The up down values are used when and insertion or deletion is detected , and if there is a match the diagonal value is taken . The min value of insertion , deletion, substitutions is taken

Applications of edit distance
Spelling Correction: Edit distance is used in spell checkers to suggest correct spellings by finding words with the smallest edit distance to the input word.

DNA Sequence Alignment: In bioinformatics, edit distance helps align DNA or protein sequences to identify similarities and differences.

Machine Translation: It is employed in machine translation systems to determine the most likely translation of a sentence by finding the edit distance between possible translations.

Text Comparison: Edit distance can be used for plagiarism detection and similarity analysis by measuring how much two pieces of text differ.

The output i got using the nucleotide sequences in the fasta file is (I used accession numbers to get the strings in the fasta file)

```
s_accession_number = "NM_001301717"
t_accession_number = "NM_001001755"

print("The edit distance between s and t is ", edit_dist(s_accession_number,t_accession_number))
```

```
The edit distance between s and t is  1880
```

3) In this question we are asked to find the global alignment between two sequences in fasta format which we have used by using accession IDs instead of writing the entire scripts Global alignment is a sequence alignment method that seeks to align the entire length of two sequences, including gaps, in a way that maximizes their overall similarity.

We have used a dynamic programming to align the sequences and compute the final result using dashes to show gaps in the sequence as shown in the output got by us , and added optimal gap and mismatch penalties . to get the most optimal alignment with the most overlap

So the scoring scheme we have used is Match-0 , mismatch- 1, gap(insertions/deletions) - 1

This is the default scoring scheme we have used in reality we are finding the optimal alignment by observing minimum edit distance between the two sequences

Some applications of global alignment
Homology Search: It helps identify homologous genes or proteins across different species to infer evolutionary relationships.
Protein Structure Prediction: Global alignment is used to identify structural similarities between proteins, aiding in the prediction of protein 3D structures.

Genome Comparison: It enables the comparison of entire genomes to study genetic variations, gene conservation, and evolutionary events.

Functional Annotation: By aligning sequences, researchers can transfer functional annotations from well-characterized sequences to less-studied ones

sample sequence which is given in the problem:
Code for the sample input:

```python
def global_alignment(s, t):
    n, m = len(s), len(t)
    dp = [[0] * (m + 1) for _ in range(n + 1)]

    # Initialize the base cases
    for i in range(n + 1):
        dp[i][0] = i
    for j in range(m + 1):
        dp[0][j] = j

    # Fill the matrix
    for i in range(1, n + 1):
        for j in range(1, m + 1):
            if s[i-1] == t[j-1]:
                match = dp[i-1][j-1]  # 0 penalty for exact matches
            else:
                match = dp[i-1][j-1] + 1  # 1 penalty for mismatches
            delete = dp[i-1][j] + 1
            insert = dp[i][j-1] + 1
            dp[i][j] = min(match, delete, insert)
```

```
    # Reconstruct the alignment
    aligned_s, aligned_t = "", ""
    i, j = n, m
    while i > 0 or j > 0:
        if i > 0 and j > 0 and (s[i-1] == t[j-1] or dp[i][j] == dp[i-1][j-1] + 1):
            aligned_s = s[i-1] + aligned_s
            aligned_t = t[j-1] + aligned_t
            i -= 1
            j -= 1
        elif i > 0 and dp[i][j] == dp[i-1][j] + 1:
            aligned_s = s[i-1] + aligned_s
            aligned_t = "-" + aligned_t
            i -= 1
        else:
            aligned_s = "-" + aligned_s
            aligned_t = t[j-1] + aligned_t
            j -= 1

    return dp[n][m], aligned_s, aligned_t

s = "PRETTY"
t = "PRTTEIN"
distance, s_prime, t_prime = global_alignment(s, t)
print(distance)
print(s_prime)
print(t_prime)
```

Output:
Edit distance is: 4.

```
4
PRET-TY
PRTTEIN
```

Output of the nucleotides which is downloaded from the ncbi fasta file:
Output:
Length of the nucleotide sequence s is 2191
Length of the nucleotide sequence t is 3537
distance comes out 1112.
We have attached only some portions of the output because output is two long

```
length of the nucleotide sequence s: 2191
length of the nucleotide sequence t: 3537
Distance: 1112
s_prime: ------------------------------------------------------------------------------
t_prime: ATGCTGCAGAGGAGCAGGCTGTTGTGGCTTGCTGTCTTCATAACCCTGTGGGTTTCCTCAGATGCTCAGGATGAT
```

```
CAAGATGAGGTCACGGACGATTACATCGGAGACAACACCACAGTGGACTACACTTTGTTCGAGTC-TTT--GTGCTCC-AAGAAGGACGTGCGGAACTTTAAAGCCTGGTTCCTCCCTATCATGTACTCCATCATTTGTTTCGTGGGCCTACTGGGCAATGGGCTGGTCGTGTTGACCTATATCT
CAA-A-GA---CA-AGAC-A-CACAGCTGCCATAA-A--TCTGCAGAGTGCA-TTTATCTGGGTCATTTCAGTGATCCTATGTA-CAAATG-TGAA-TGT-AGGACAGGTT-ATGCTGGTGATGGAC-GCATC---TG--TGGTGAG--GATTCAG-ATTTGGATGG--ATG--G-CC-A-A-AT
```