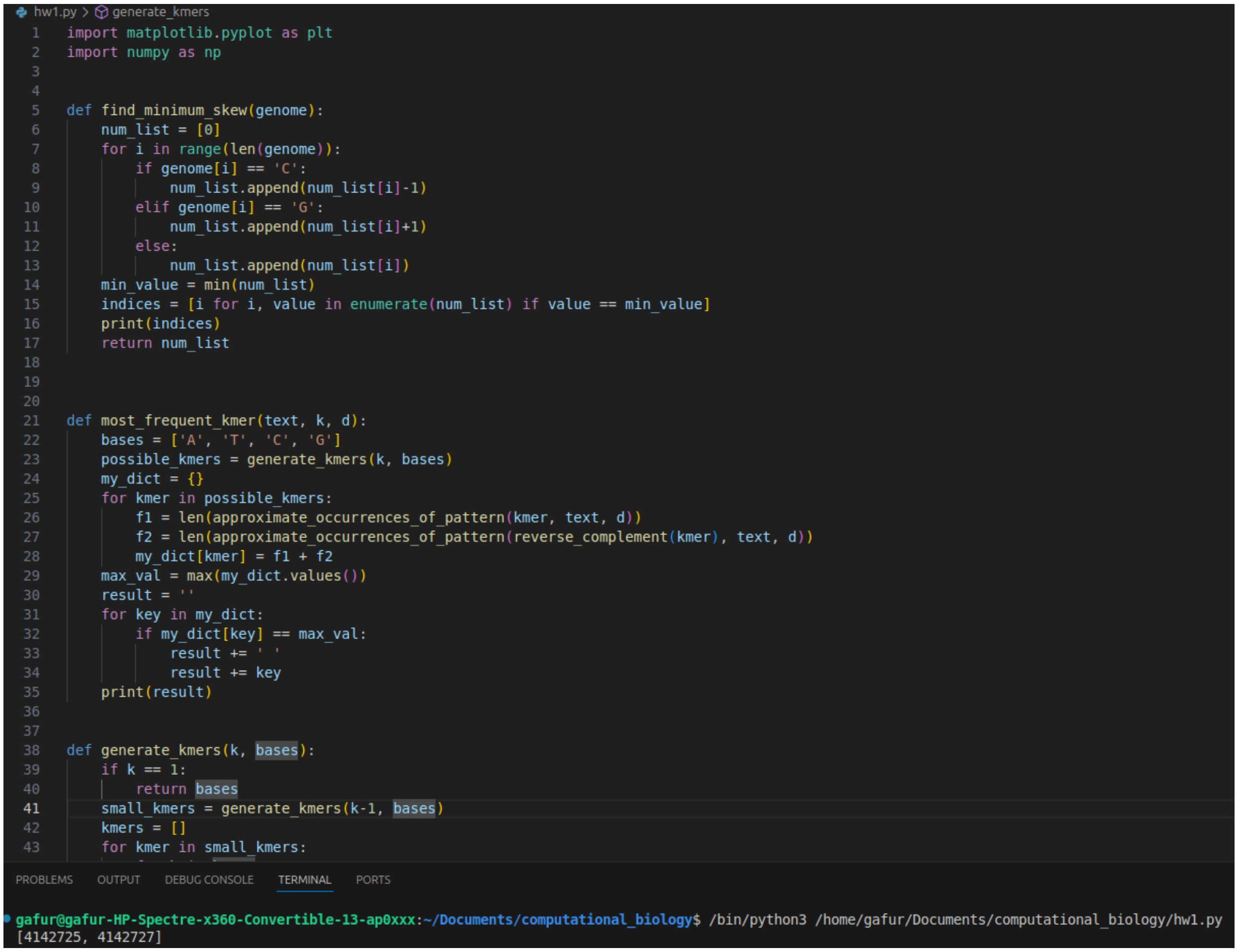
1) My first approach to finding the *ori* of Salmonella Enterica will start with finding the genome's minimum skew in the genome.

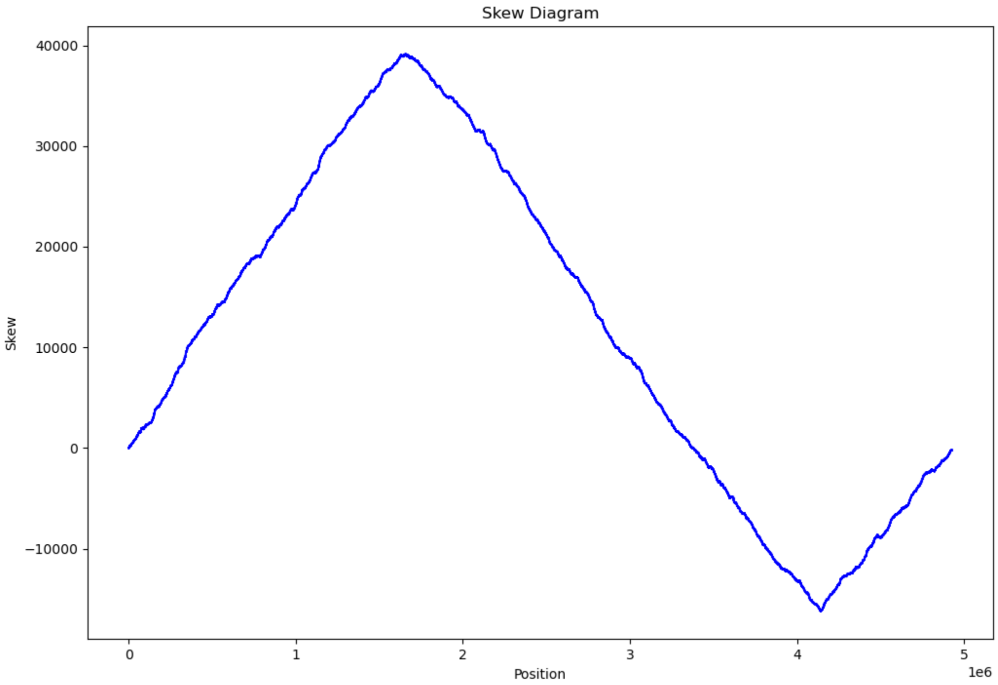
My output for the skew finding algorithm is as below:



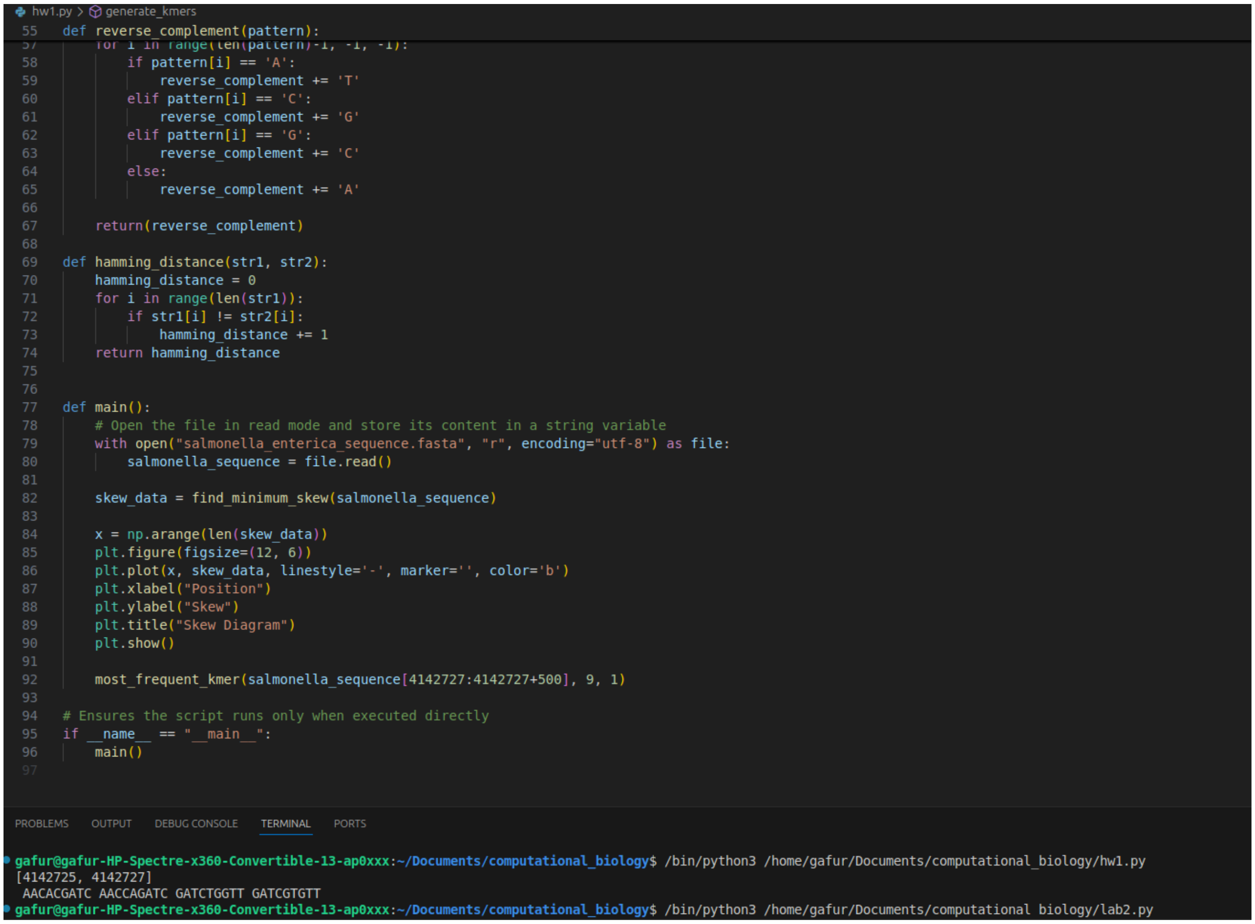
```python
import matplotlib.pyplot as plt
import numpy as np


def find_minimum_skew(genome):
    num_list = [0]
    for i in range(len(genome)):
        if genome[i] == 'C':
            num_list.append(num_list[i]-1)
        elif genome[i] == 'G':
            num_list.append(num_list[i]+1)
        else:
            num_list.append(num_list[i])
    min_value = min(num_list)
    indices = [i for i, value in enumerate(num_list) if value == min_value]
    print(indices)
    return num_list



def most_frequent_kmer(text, k, d):
    bases = ['A', 'T', 'C', 'G']
    possible_kmers = generate_kmers(k, bases)
    my_dict = {}
    for kmer in possible_kmers:
        f1 = len(approximate_occurrences_of_pattern(kmer, text, d))
        f2 = len(approximate_occurrences_of_pattern(reverse_complement(kmer), text, d))
        my_dict[kmer] = f1 + f2
    max_val = max(my_dict.values())
    result = ''
    for key in my_dict:
        if my_dict[key] == max_val:
            result += ' '
            result += key
    print(result)


def generate_kmers(k, bases):
    if k == 1:
        return bases
    small_kmers = generate_kmers(k-1, bases)
    kmers = []
    for kmer in small_kmers:
```

```
gafur@gafur-HP-Spectre-x360-Convertible-13-ap0xxx:~/Documents/computational_biology$ /bin/python3 /home/gafur/Documents/computational_biology/hw1.py
[4142725, 4142727]
```

This is the skew diagram that I created for better understanding. I made the conclusion that the position of minimum skew is **4142727.**



2) After finding the minimum skew position, to make sure that it is the origin of replication, I checked the most frequent 9-mer with 1 mismatch and its reverse complements.
   - I did this check in between positions [**4142725: 4142725+500].**
   - It turns out that the most frequent 9-mers are – **AACACGATC, AACCAGATC, GATCTGGTT, GATCGTGTT** (GATCTGGTT and GATCGTGTT are reverse complements of AACACGATC, AACCAGATC).
   - This is the screenshot of my output:



```python
def reverse_complement(pattern):
    for i in range(len(pattern)-1, -1, -1):
        if pattern[i] == 'A':
            reverse_complement += 'T'
        elif pattern[i] == 'C':
            reverse_complement += 'G'
        elif pattern[i] == 'G':
            reverse_complement += 'C'
        else:
            reverse_complement += 'A'

    return(reverse_complement)

def hamming_distance(str1, str2):
    hamming_distance = 0
    for i in range(len(str1)):
        if str1[i] != str2[i]:
            hamming_distance += 1
    return hamming_distance


def main():
    # Open the file in read mode and store its content in a string variable
    with open("salmonella_enterica_sequence.fasta", "r", encoding="utf-8") as file:
        salmonella_sequence = file.read()

    skew_data = find_minimum_skew(salmonella_sequence)

    x = np.arange(len(skew_data))
    plt.figure(figsize=(12, 6))
    plt.plot(x, skew_data, linestyle='-', marker='', color='b')
    plt.xlabel("Position")
    plt.ylabel("Skew")
    plt.title("Skew Diagram")
    plt.show()

    most_frequent_kmer(salmonella_sequence[4142727:4142727+500], 9, 1)

# Ensures the script runs only when executed directly
if __name__ == "__main__":
    main()
```

```
gafur@gafur-HP-Spectre-x360-Convertible-13-ap0xxx:~/Documents/computational_biology$ /bin/python3 /home/gafur/Documents/computational_biology/hw1.py
[4142725, 4142727]
 AACACGATC AACCAGATC GATCTGGTT GATCGTGTT
gafur@gafur-HP-Spectre-x360-Convertible-13-ap0xxx:~/Documents/computational_biology$ /bin/python3 /home/gafur/Documents/computational_biology/lab2.py
```

- My algorithm starts with creating all possible 9–mers and then uses dictionary to find the most frequent ones. (I did not use the neighbor technique)

3) In conclusion, I identified the positions **[4142725: 4142725+500]** as DnaA box because, first, the position of minimum skew starts from there. Secondly, there are 2 most frequent 9-mers in between these positions.

4) One drawback of my approach is that it requires more computation because it does not utilize d-neighborhood technique.