2B – Row 2 (Iterative Development Process)

My iterative development process consisted of building each function or block of code, testing it with what was written before, fixing any issues, and then either moving on to the next function or scrapping the failed one and rewriting or redesigning it completely. After completing a major feature of the program, I went back and reflected on the code, tweaking it in order to perfect it as much as possible. I rewrote some functions multiple times in order to streamline the program and make it easier to understand.

2C – Row 3 (Challenges, and how they were solved)

When developing this program, I came across significant difficulties in creating both the functions for reversing and spacing the answer string. I had to reverse the answer string to account for DNA prime ends, and I spent a considerable amount of time trying workarounds, until I eventually discovered the reversed() function, which does the entire job in one line. For spacing, I spent multiple hours trying various complicated workarounds using for loops and other ideas, which turned out to be impossible without hardcoding. The solution was to use the join() function in conjunction with a for loop to insert spaces into the string.

2D – Row 4 (Algorithm)

```
def translate(dna):
    toFlip = find_complimentary_bases(dna)
    final = flip_strand(toFlip)
    final = space_strand(final)
    return final
```

This algorithm, translate, is responsible for calling several other algorithms (defined in separate functions) that work in order to translate and format the answer requested by the user. The first algorithm takes the users input and returns the answer as a list, with each index containing one character. The next algorithm takes the previous list and reverses it's order. The third function takes the reversed list, converts it into a string, inserts spaces at regular intervals, and returns it to translate, which then returns it to the original function call, printing the answer.