Code 1.py

import argparse

def hide\_message(doc\_file, secret\_msg, output):

with open(doc\_file, 'r') as file:

content = file.read()

content += "\n" + secret\_msg # No need for delimiters in DNA steganography

with open(output, 'w') as file:

file.write(content)

print("Secret message hidden successfully.")

def main():

parser = argparse.ArgumentParser()

parser.add\_argument('-f', help='DNA Sequence File', dest='dnafile', required=True)

parser.add\_argument('-m', help='Enter your Secret Message', dest='secretmsg', required=True)

parser.add\_argument('-o', help='Output File Path and Name', dest='outputfile', required=True)

args = parser.parse\_args()

dna\_file = args.dnafile

secret\_msg = args.secretmsg

output = args.outputfile

try:

hide\_message(dna\_file, secret\_msg, output)

except FileNotFoundError:

print("File not found. Please provide a valid file path.")

except Exception as e:

print("An error occurred:", e)

if \_\_name\_\_ == "\_\_main\_\_":

main()

code2.py

import argparse

parser = argparse.ArgumentParser()

parser.add\_argument('-f', help='DNA Sequence File', dest='dnafile')

args = parser.parse\_args()

dna\_file = args.dnafile

arged = False

if dna\_file:

arged = True

def extract\_message(input\_dna\_file):

if not arged:

print("Usage: python ExtractDNA.py -f [DNA Sequence File]")

else:

try:

with open(input\_dna\_file, 'r') as file:

content = file.read()

# In DNA steganography, secret message is directly embedded in the sequence

print("Extracted secret message:", content.strip())

except FileNotFoundError as file\_err:

print("File not found:", file\_err)

except Exception as other\_err:

print("An error occurred:", other\_err)

try:

extract\_message(dna\_file)

except KeyboardInterrupt:

print("Keyboard interrupt detected. Exiting...")

except Exception as ex:

print("Something went wrong! Please try again:", ex)