**Pregunta 1:**

**Escribe un código en Python que reciba una cadena de ADN como entrada y verifique si comienza con el codón de inicio “ATG” y termina con alguno de los codones de stop: “TAA”, “TAG” o “TGA”.**

def verificar\_codon\_inicio\_y\_stop(ADN):

ADN = ADN.upper().replace(" ", "")

if ADN[:3] == "ATG" and ADN[-3:] in ["TAA", "TAG", "TGA"]:

return True

else:

return False

ADN = input("Introduce la secuencia de ADN: ")

if verificar\_codon\_inicio\_y\_stop(ADN):

print("La cadena de ADN tiene el codón de inicio 'ATG'.")

**Pregunta 2:**

**Escribe una función traducir codones que, dada una cadena de ARN válida (que comienza en AUG y cuya longitud sea múltiplo de 3), divida la cadena de 3 en 3 (en codones) y los muestre como una lista.**

def traducir\_codones(ARN):

if len(ARN) % 3 != 0:

return "No es valida."

codones = [ARN[i:i+3] for i in range(0, len(ARN), 3)]

return codones

ARN = "AUGGCCUUUGGA"

resultado = traducir\_codones(ARN)

print(“Lista de codones:”resultado)

**Pregunta 3:**

**Define una función que reciba una lista de cadenas de ADN y retorne una lista con los complementos de cada una.**

complementos = {'A': 'T', 'T': 'A', 'C': 'G', 'G': 'C'}

ADN = input("Introduce la secuencia de ADN: ")

ADN = ADN.upper().replace(" ", "")

ADNc = ""

for letra in ADN:

if letra in complementos:

ADNc += complementos[letra]

if ADNc:

print("Secuencia original:", ADN)

print("Complemento:", ADNc)

**Pregunta 4:**

**Dada una lista de cadenas de ADN, escribe una función que:**

* **Calcule su complemento.**
* **Solo retorne las cadenas complementarias que empiecen con “ATG”**

ADN = "ATGCGTAGCTAGCTAGCTAGCTAGC"

def complemento(ADN):

complemento\_dict = {'A': 'T', 'T': 'A', 'C': 'G', 'G': 'C'}

ADN = ADN.upper().replace(" ", "")

ADNc = ""

for letra in ADN:

ADNc += complemento\_dict[letra]

return ADNc

def complemento\_con\_inicio\_ATG(ADN):

ADNc = complemento(ADN)

if ADNc[:3] == "ATG":

return ADNc

return None

print(complemento\_con\_inicio\_ATG("attggggaaatt tt"))

**Pregunta 5:**

**Crea un código que, dada una cadena larga de ARNm, busque todas las posiciones donde aparece el codón “AUG” y, desde cada una, intente traducir hasta un codón de parada (dada por el usuario). Imprime todas las secuencias traducidas encontradas.**

def traducir\_ ARNm (ARNm, codon\_parada):

ARNm = ARNm.upper().replace(" ", "")

codon\_parada = codon\_parada.upper()

secuencias\_traducidas = []

for i in range(len(ARNm) - 2)

if ARNm [i:i+3] == "AUG":

secuencia = ""

for j in range(i, len(ARNm), 3):

codon = arnm[j:j+3]

if len(codon) < 3:

break

if codon == codon\_parada: secuencias\_traducidas.append(secuencia)

break

secuencia += codon

return secuencias\_traducidas

ARNm = "AUGGCCUUUGGAUGCCCUAAUGGGAUUUUGA"

codon\_parada = input("Introduce el codón de parada")

secuencias = traducir\_ ARNm (ARNm, codon\_parada)

if secuencias:

print("Secuencias traducidas encontradas:")

for secuencia in secuencias:

print(secuencia)