def hamming\_distance(str1="ADEDAYO", str2="ARA\_AAT"):

if len(str1) != len(str2):

raise ValueError("Sequences must be of equal length.")

return sum(a != b for a, b in zip(str1, str2))

print(hamming\_distance()) # Uses default values

def dna\_to\_protein(dna\_sequence):

codon\_table = {

"ATA": "I", "ATC": "I", "ATT": "I", "ATG": "M",

"ACA": "T", "ACC": "T", "ACG": "T", "ACT": "T",

"AAC": "N", "AAT": "N", "AAA": "K", "AAG": "K",

"AGC": "S", "AGT": "S", "AGA": "R", "AGG": "R",

"CTA": "L", "CTC": "L", "CTG": "L", "CTT": "L",

"CCA": "P", "CCC": "P", "CCG": "P", "CCT": "P",

"CAC": "H", "CAT": "H", "CAA": "Q", "CAG": "Q",

"CGA": "R", "CGC": "R", "CGG": "R", "CGT": "R",

"GTA": "V", "GTC": "V", "GTG": "V", "GTT": "V",

"GCA": "A", "GCC": "A", "GCG": "A", "GCT": "A",

"GAC": "D", "GAT": "D", "GAA": "E", "GAG": "E",

"GGA": "G", "GGC": "G", "GGG": "G", "GGT": "G",

"TCA": "S", "TCC": "S", "TCG": "S", "TCT": "S",

"TTC": "F", "TTT": "F", "TTA": "L", "TTG": "L",

"TAC": "Y", "TAT": "Y", "TAA": "Stop", "TAG": "Stop",

"TGC": "C", "TGT": "C", "TGA": "Stop", "TGG": "W"

}

protein = ""

for i in range(0, len(dna\_sequence) - len(dna\_sequence) % 3, 3):

codon = dna\_sequence[i:i+3]

amino\_acid = codon\_table.get(codon, "?") # Use "?" for unknown codons

if amino\_acid == "Stop":

break # Stop translation at stop codon

protein += amino\_acid

return protein # Returns translated protein sequence

# Example usage

dna\_sequence = "ATGGCTTAG"

protein = dna\_to\_protein(dna\_sequence)

print("Translated Protein:", protein)