Write a C program for possible keys does the Playfair cipher have? Ignore the fact that some keys might produce identical encryption results. Express your answer as an approximate power of 2. a. Now take into account the fact that some Playfair keys produce the same encryption results. How many effectively unique keys does the Playfair cipher have?

Program:

import math

# Function to compute log2(n!)

def log2\_factorial(n):

log2\_sum = 0

for i in range(2, n+1):

log2\_sum += math.log2(i)

return log2\_sum

# Total letters in Playfair (I/J combined)

n = 25

# Total possible keys ignoring duplicates

total\_keys\_log2 = log2\_factorial(n)

# Effectively unique keys (approx.) accounting for row/column symmetries

# Dividing by 5! for rows and 5! for columns (approximation)

effective\_keys\_log2 = total\_keys\_log2 - 2\*log2\_factorial(5)

print("Playfair Cipher Key Analysis")

print("----------------------------")

print(f"1. Total possible keys (ignoring duplicates): 2^{total\_keys\_log2:.0f}")

print(f"2. Effectively unique keys (considering symmetries): 2^{effective\_keys\_log2:.0f}")

output:

