

PYTHON CODE AS RAN:

```
import math
from decimal import Decimal

sequence_length = 5      # Length of the sequence to be investigated
identity = 60             # Minimum % identity of sequences to consider

# Calculate how many bases (positions) within the sequence need to be changed for the % identity entered above
n = sequence_length
positions = int(n*((100-identity)/100))

number_of_sequences = 0   # This is where we will store the answer
k = 1                     # We will start with 1 position and then sum over all the positions we need as defined above

# The while loop sums from 1 to the number of positions calculated above
while k <= positions:
    # math.comb calculates  $n!/(k!(n-k)!)$  which is the number of ways to choose k items from n items without repetition
    # and without order
    # in our case n = sequence_length and k is the position
    # we also multiply by  $3^k$  because at each position there are three possible sequence changes that result in a
    # mutation
    number_of_sequences += Decimal(math.comb(n, k)*(3**k))
    k += 1

print("The total number of sequences that are at least " + str(identity) + "% identical to a " + str(n) + "-nucleotide
sequence is " + "{:.2e}".format(number_of_sequences))
```

OUTPUT FROM THIS CODE:

The total number of sequences that are at least 60% identical to a 5-nucleotide sequence is
1.05e+2

NOTES:

1.05e+2 is scientific notation for 105