1 Reverse complement

In the DNA double helix, two strands twist together and “face” each other. The two strands are reverse-
complementary, i.e., reading one strand in reverse order and exchanging each base with its complement gives the
other strand. A and T are complementary; C and G are complementary.

For example, given the DNA sequence

AGTACAT

the reverse sequence is

TACGATGA

so the reverse complement is

ATGCTACT

(a) Write a function \( \text{rComplement}(\text{dna}) \) to return the reverse complement of a DNA strand. Use a loop to reverse
the strand—do not use vectorized code. \( \text{dna} \) is a vector of characters. Assume that \( \text{dna} \) contains only the letters
'A', 'T', 'C', and 'G'. If \( \text{dna} \) is the empty vector return the empty vector.

(b) Now write a function \( \text{rCompBulk}(\text{mat}) \) to return the reverse complements of a set of DNA strands. \( \text{mat} \) is
a matrix of characters; each row of the matrix represents one strand of DNA (so \( \text{mat} \) contains only the letters
'A', 'T', 'C', and 'G'). Return a matrix the same size as \( \text{mat} \) such that the \( r \)th row of the returned matrix is
the reverse complement of the \( r \)th strand of DNA (the \( r \)th row of \( \text{mat} \)). Again use loops—do not use vectorized
code.

2 Counting a DNA pattern

Write a function \( \text{countPattern}(\text{dna}, \text{p}) \) to find out (and return) how many times a pattern \( \text{p} \) occurs in
\( \text{dna} \). Assume both parameters to be strings that contain the letters 'A', 'T', 'C', and 'G' only. Note that if \( \text{p} \) is longer
than \( \text{dna} \), then \( \text{p} \) appears in \( \text{dna} \) zero times. Use the built-in function \( \text{strcmp} \) to compare two strings. Again,
use a loop to solve this problem.

3 Counting a DNA pattern—challenge edition!

Rewrite the function \( \text{countPattern}(\text{dna}, \text{p}) \) without using the \( \text{strcmp} \) built-in function.