

# Longest palindrome

A DNA sequence can be represented by a string that consists solely of the letters A, C, G and T. The *inverse complement* of a DNA sequence is reached by reversing the order of the letters in the string and changing every A into a T (and the other way around) and changing every G into a C (and the other way around). The inverse complement of the DNA sequence AGCTGCT, for example, turns into AGCAGCT. A *palindromic sequence* is a DNA sequence that reads exactly the same on the complementary string. The inverse complement of the DNA sequence AATGCATT, for example, equals the original DNA sequence. This is a palindromic sequence.

## Assignment

1. Write a function `DNAPalindrome` to which a DNA sequence should be given as an argument. As a result, the function should print the value `True` if the DNA sequence is a palindromic sequence. Otherwise, the value `False` should be printed. For example, the function should print `True` for the DNA sequence `GAATTC`.
2. Use the function `DNAPalindrome` to write the function `longestPalindrome`, to which a DNA sequence should be given as an argument. As a result, the function should give the longest palindromic partial sequence of the sequence given. To find the longest palindromic partial sequence, you can go through every partial sequence possible and test whether or not you are dealing with a palindromic sequence and eventually be left with the longest palindrome. In order to run through all partial sequences of a given DNA sequence, you can run through all starting positions possible and for every starting position, go through the possible ending positions. This way, you will find every partial sequence as the sequence between every combination of possible starting and ending positions.

## Example

```
>>> DNAPalindrome('GAATTC')
True
>>> DNAPalindrome('AGACTCT')
False
>>> longestPalindrome('CCCCCCCCGAATTCTTTTATTTT')
'GAATTC'
>>> longestPalindrome('')
''
```

Een DNA sequentie kan voorgesteld worden door een string die enkel bestaat uit de letters A, C, G en T. Het *invers complement* van een DNA sequentie wordt bekomen door de volgorde van de letters in de string om te keren, en daarbij elk voorkomen van een A te wijzigen in een T (en omgekeerd) en elk voorkomen van een G in een C (en omgekeerd). Zo is het invers complement van de DNA sequentie AGCTGCT bijvoorbeeld AGCAGCT. Een *palindromische sequentie* is een DNA sequentie die precies hetzelfde leest op de complementaire streng. Zo is het invers complement van de DNA sequentie AATGCATT gelijk aan de oorspronkelijke DNA sequentie, en is deze DNA sequentie dus een voorbeeld van een palindromische sequentie.

## Opgave

1. Schrijf een functie `DNAPalindroom` waaraan een DNA sequentie als argument moet

doorgegeven worden. Deze functie moet als resultaat de waarde `True` teruggeven indien de gegeven DNA sequentie een palindromische sequentie is. Anders moet de waarde `False` teruggegeven worden. Zo moet de functie voor de DNA sequentie `GAATTC` de waarde `True` teruggeven.

2. Gebruik de functie `DNAPalindroom` om een functie `langstePalindroom` te schrijven. Aan deze functie moet een DNA sequentie als argument doorgegeven worden. De functie moet de langste palindromische deelsequentie van de gegeven sequentie als resultaat teruggeven. Om de langste palindromische deelsequentie te vinden, kan je alle mogelijke deelsequenties van de gegeven DNA sequentie overlopen, testen of het gaat om een palindromische sequentie, en uiteindelijk het langste palindroom overhouden. Om alle mogelijke deelsequenties van een gegeven DNA sequentie te doorlopen, kan je alle mogelijke startposities doorlopen en voor elke mogelijke startpositie alle mogelijke stopposities doorlopen. Op die manier vind je elke deelsequentie als de sequentie tussen elke combinatie van mogelijke start- en stopposities.

## Voorbeeld

```
>>> DNAPalindroom('GAATTC')
True
>>> DNAPalindroom('AGACTCT')
False
>>> langstePalindroom('CCCCCCCCGAATTCTTTTATTTT')
'GAATTC'
>>> langstePalindroom("")
''
```