**Universität Potsdam**
**Institut für Informatik**

GdP-Rechnerübung

**Aufgabenblatt 14**

**Lernziele (zum Abhaken):** Die Student:innen können...
☐ einen vorgegeben Algorithmus unter der Benutzung der rekursiven Programmierung implementieren.
☐ ein vorgegebenes Testszenario für die Überprüfung der Funktionalität eines Programmes implementieren.
☐ einen vorgegeben Algorithmus unter der Verwendung der funktionalen Programmierung implementieren.

## 26 Rekursive Programmierung in Python

1. Implementieren Sie die folgenden Funktionen rekursiv und schreiben Sie eine Testanwendung, die die Funktionsweise Ihrer Implementationen überprüft:

   a) **Funktion:** Implementieren Sie die Multiplikation zweier natürlicher Zahlen, die nur auf der Addition basiert.

   **Testanwendung:** Die Testanwendung fordert den Benutzer zur Eingabe zweier natürlicher Zahlen auf und berechnet deren Produkt durch Aufruf der rekursiven Funktion. Sie soll eine aussagekräftige Fehlermeldung ausgeben, falls negative Werte eingegeben werden.

   b) **Funktion:** Implementieren Sie eine Funktion, die die Länge einer Liste berechnet. (Benutzen Sie `map` nicht.)

   **Testanwendung:** Die Testanwendung ruft die Funktion mit einer beliebigen (nicht vom Nutzer eingegebenen) Liste auf.

   c) **Funktion:** Implementieren Sie die Fibonacci-Funktion aus der Übung 7.

   **Testanwendung:** Die Testanwendung fordert den Benutzer zur Eingabe einer natürlichen Zahl $n$ auf und gibt daraufhin die ersten $n$ Fibonacci-Zahlen aus. Sie soll eine aussagekräftige Fehlermeldung ausgeben, falls eine negative Zahl eingegeben wird.

## 27 Funktionale Programmierung in Python

1. Benutzen Sie Pythons Funktionen `map, reduce, filter`, um die folgenden Funktionen auf Listen zu implementieren.

   a) Alle Werte einer Liste werden auf -1 gesetzt.

   b) Alle geraden Zahlen werden aus der Liste entfernt.

   c) Die Länge einer Liste wird errechnet.

2. Strings kann man in Python wie Listen behandeln. Schreiben Sie eine Funktion, die hinter jedes Zeichen ein Leerzeichen einfügt.

   Beispiel: "Hallo Welt"→ "H a l l o W e l t "

3. Implementieren Sie die in der Vorlesung und Übung kennengelernten Funktionalitäten von Linkssequenzen mittels Tupeln in Python. Definieren Sie am Anfang Ihres Python-Programms `empty=()`, um die in der Vorlesung vorgestellte Schreibweise benutzen zu können.

   Schreiben Sie die folgenden Funktionen:

   a) `first(xs)`, die das erste Element der Linkssequenz zurückgibt.

   b) `rest(xs)`, die den Rest der Linkssequenz zurückgibt.

      Benutzen Sie ab jetzt nur noch `first` und `rest`, um auf die Linkssequenzen zuzugreifen!

   c) `fromLtoLseq(L)`, die eine Liste in eine Linkssequenz umwandelt.

   d) `lastElement(xs)`, die das letzte Element der Linkssequenz zurückgibt.

   e) `nElement(n, xs)`, die das $n$te Element liefert.

   f) `concat(xs, ys)`, die xs mit ys konkateniert.

   g) `init(xs)`, die die Linkssequenz bis auf das letzte Element liefert.

   h) `reverse(xs)`, die die Linkssequenz mit umgekehrter Reihenfolge ihrer Elemente zurückgibt.

   i) `ls2rs(xs)`, die zu einer Linkssequenz eine Rechtssequenz mit den gleichen Elementen liefert.

4. Schreiben Sie jetzt eine Variante von `map` für Linkssequenzen.

5. Verwenden Sie Ihre Funktion `map`, um das folgende Verhalten zu implementieren:

   a) Die Werte aller Elemente der Linkssequenz werden verdoppelt.

   b) Alle Werte der Linkssequenz werden auf 0 gesetzt.

   c) Anstelle jedes Elements wird die Länge seiner String-Repräsentation geschrieben.

## Zusatzaufgabe

1. Schreiben Sie eine Funktion, die aus einem String eine Linkssequenz aus Zeichen erstellt.

2. Schreiben Sie eine Funktion, die erkennt, ob es sich bei dem String in Linkssequenz-Repräsentation um ein Palindrom handelt.

   *Hinweis:* Palindrome sind Zeichenfolgen, die vorwärts- sowie rückwärts gelesen gleich sind – zum Beispiel „ANNA“.