

Übungsblatt 8

(Decade I, Decadi de Frimaire de l'Annee 218 de la Revolution)

Aufgabe 1 (1 Punkt) Gegeben sei die folgende Headerdatei für eine kleine Bibliothek zum Arbeiten mit Zeichenketten:

```
1 typedef enum {false,true} boolean;
2 typedef char* string;
3 typedef string wordlist [];
4 typedef unsigned int nat;
5
6 boolean isVowel(char c);
7 nat strLength(string this);
8 boolean strEq(string this, string that);
9 boolean doRhyme(string this, string that);
10 string reverse(string this);
11 boolean isSmallerEq(string this, string that);
12 boolean isRevertedSmallerEq(string this, string that);
```

Implementieren Sie die Bibliothek ohne Zuhilfenahme der Standardstringbibliothek. Testen Sie Ihre Implementierung mit dem Programm `TestStringUtil.c`

- a) `isVowel` soll testen, ob das übergebene Zeichen ein Vokal ist.
- b) `strLength` soll die Länge des übergebenen Strings zurückgeben.
- c) `strEq` soll testen, ob es sich um gleiche Zeichenketten handelt.
- d) `reverse` soll einen neuen String erzeugen und zurückgeben, der gerade aus der umgekehrten Zeichenkette besteht.
- e) `doRhyme` soll testen, ob die beiden übergebenen Wörter sich reimen. Hierzu müssen die Wörter sich vom ersten Vokal an gleichen.

Aufgabe 2 Implementieren Sie die verbleibenden zwei Funktionen.

- a) `isSmallerEq` soll testen ob in der lexikographischen Ordnung (also nach dem Alphabet »abcdefghijklmnopqrstuvwxyz«) das erste Argument früher kommt als das zweite.
- b) `isReverseSmallerEq` soll schauen, ob das erste Wort in der lexikographischen Ordnung rückwärtsgelesen vor dem zweiten Wort kommt.