

Betreutes Programmieren 9

Auf diesem Übungsblatt werden Sie sich mit Zeigern und der Umwandlung von Zeichenketten in Zahlen beschäftigen.

Aufgabe 9 (*Umwandlung von Zeichenketten in Kommazahlen*)

a) Schreiben Sie eine Funktion mit dem Prototypen

```
double string_to_double(const char *s, char **endp),
```

die sich wie die Bibliotheksfunktion `strtod` verhält. Fehlerfälle (Werte, bei denen es zu einem Overflow oder Underflow kommt), müssen Sie dabei nicht berücksichtigen. Sie können also davon ausgehen, dass die umgewandelte Zahl als `double` dargestellt werden kann, sofern die Zeichenkette eine Zahl enthält. Sie dürfen außerdem annehmen, dass die umzuwandelnde Zahl in Festkommenschreibweise gegeben ist.

b) Testen Sie Ihre Umwandlungsfunktion mithilfe des vorgegebenen Hauptprogramms. Die Aufrufe der `test`-Funktion sollten dazu führen, dass immer zweimal das gleiche Ergebnis ausgegeben wird (einmal von der Bibliotheksfunktion berechnet und einmal von der eigenen Funktion), sofern Ihre Umwandlungsfunktion richtig funktioniert. Falls für eine Eingabe unterschiedliche Ergebnisse ausgegeben werden, überlegen Sie woran das liegt und ergänzen Sie Ihre `string_to_double`-Funktion entsprechend, um die Umwandlung für ähnliche Fälle korrekt durchzuführen.

c) *Freiwillige Zusatzaufgabe:* Erweitern Sie Ihre `string_to_double`-Funktion so, dass auch Zahlen in Gleitkommadarstellung korrekt umgewandelt werden. Ergänzen Sie zum Testen der erweiterten Funktion folgende Aufrufe in der `main`-Funktion:

```
test("65e7", endp1, endp2);
test("786e-6", endp1, endp2);
test(" 43e11", endp1, endp2);
test("+4387e2", endp1, endp2);
test("-432e4", endp1, endp2);
test("-167e-3", endp1, endp2);
test("6538.3e2Ab", endp1, endp2);
test("175e", endp1, endp2);
test("175e-", endp1, endp2);
```