

Auszuarbeiten bis 21.01.20

1. Kleinstes gemeinsames Vielfaches (4 Punkte)

Entwickeln Sie eine Funktion, die das kleinste gemeinsame Vielfache zweier ganzer Zahlen bestimmt und als Ergebnis zurückliefert.

2. Modularisierung (16 Punkte)

Prof. Mayar ist ganz zappelig: Seit Monaten möchte er ein Archivierungsprogramm schreiben, doch alle seine Versuche scheiterten, weil der Code zu unübersichtlich wurde. In der vergangenen schlaflosen Nacht fiel ihm plötzlich das Konzept der Modularisierung ein. Nun hat er sich eifrig darangemacht, eine Schnittstelle zu definieren und an seinem Programm weiterzuschreiben.

Helfen Sie Professor Mayar dabei, sein Programm zu vervollständigen, indem Sie die benötigten *Stringfunktionen* in einem eigenen Modul zur Verfügung stellen.

Das Modul sollte folgende Funktionen beinhalten:

- `int string_len(const char[] str)`
 - liefert die Länge eines Strings zurück
- `int string_cmp(const char[] str1, const char[] str2)`
 - liefert 0 zurück, wenn die beiden Strings identisch sind, -1 wenn `str1 < str 2`, +1 wenn `str1 > str2`
 - hierbei zählt der Buchstabenweise Vergleich, nicht die Stringlänge (diese zählt nur bei gleichen Strings),
z.B.: „Anton“ < „Antonius“ < „Armin“ < „Berta“ < „Bertram“
- `void string_toUpperCase(char[] str)`
 - Wandelt einen String in Großbuchstaben um
 - Anmerkung: Es kann direkt auf dem als Parameter übergebenen String gearbeitet werden
- `void string_copy(char[] destStr, const char[] srcStr)`
 - kopiert den String `srcStr` nach `destStr`
 - Anmerkung: Es kann direkt in den als Parameter übergebenen String kopiert werden (für diesen muss genügend Platz reserviert sein)
- `int string_subStr(char[] destStr, const char[] srcStr, int startOffs, int endOffs)`
 - kopiert den String `srcStr` von der Stelle `startOffs` bis zur Stelle `endOffs` nach `destStr`
 - gibt die Anzahl der kopierten Zeichen zurück, wenn die Funktion erfolgreich war, sonst -1 (z.B. wenn ein Offset hinter das Stringende von `srcStr` zeigt)
- `int string_revert(char[] str)`
 - Invertiert eine Zeichenkette.

Schreiben Sie für dieses Programm eine Header (`.h`)- und zwei Quelltextdateien (`.c`):

- **`stringfunc.c`** enthält die Stringfunktionen
- **`stringfunc.h`** enthält Prototypen für die Funktionen aus `stringfunc.c`, die vom Hauptprogramm aus, gebraucht werden
- **`main.c`** enthält **ausführliche Tests** für **alle** implementierten Funktionen.

Verwenden Sie bei der Implementierung keine String-Bibliotheksfunktionen!