Strukturierte Programmierung Übung 9

Auszuarbeiten bis 21.01.20

1. Kleinstes gemeinsames Vielfaches (4 Punkte)

Entwickeln Sie eine Funktion, die das kleinste gemeinsame Vielfacher 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 einen 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!

1 Kleinstes gemeinsames Vielfaches

Lösungsidee:
Das kleinste gemeinsame Vielfache wird folgend berechnet:
num1 * num2 / GCT (greatest common divisor von num1 und num 2)
Nur positive Zahlen sind erlaubt.

Testfälle:
(10, 10); (10, 4); (170, -60); (23, 0)

C-Program code: Siehe beigelegtes leastCommonMultiple.c/.h file.
Tests: Siehe beigelegtes main.c file.

2 Modularisierung

Lösungsidee:
Schnittstelle für folgende string Functionen schreiben.

string_len

Solange kein Terminierungszeichen auftritt durch das char Array iterieren und einen Counter bei jeder Iteration um 1 hinaufzählen.

Testfälle:
"TestThisLength", "", "1"

string_cmp

Hier wird buchstabenweise überprüft ob das aktuelle Zeichen größer, kleiner oder gleich ist. Siehe Angabe.

Testfälle:
("TestString", "TestString"); ("Test", "Tasten"); ("Tast", "Testen"); ("Test", "TestString");
("TestString", "Test"); ("A", "B"); ("B", "A"); ("", "")

string_toUpperCase

String Länge ermitteln und beim der Iteration auf einen kleinen Buchstaben überprüfen, falls einer existiert wird mit folgender Berechnung der Buchstabe groß gemacht und überschrieben: „str[i] - 'a' + 'A'“

Testfälle:
"TestString", "", "56asd", " "

string_copy
Zuerst genug Speicher für den destination String reservieren, dann Zeichen für Zeichen vom source String hinüber kopieren. Am Ende noch das Terminierungszeichen beim destination String hinzufügen.

Testfälle: (der zu kopierende string)
"TestString", "42 Hahaa", " ", ""

string_subStr
Hier wird, mit einem start Offset und einem end Offset, der destination String vom source String hinausgeschnitten. Dabei muss man beachten, dass die Offsets passen d.h. end Offset darf nicht kleiner sein als start Offset, beide Offsets dürfen nicht größer sein als der source String selber und der delta Offset (endOffs - startOffs) muss >= 0 sein. Zurückgegeben wird die anzahl der kopierten Zeichen und bei fehlschlag der Funktion, wegen den bevor angegebenen Gründen, wird -1 zurückgegeben.

Testfälle:
("TestString", 2, 5); ("TestString", 0, 9); ("TestString", 0, 15); ("TestString", -3, 9); ("TestString", 6, 3); ("TestString", 0, 0); ("TestString", 5, 5);

string_revert
Hier wird einfach, indem man das letzte Zeichen (j = length of string – 1 -> wird bei jeder Iteration dekrementiert) und das erste Zeichen (i = 0 -> wird bei jeder Iteration inkrementiert) miteinander tauscht, der String reverted. Dabei wird das gemacht solange i < j ist.

Testfälle:
"RevertThisString", "Revert This String", "username: xdorphine56", "", " "

C-Program code: Siehe beigelegtes stringfunc.c/.h file.
Tests: Siehe beigelegtes main.c file.