

Übung Funktionen Erstellen

Inhaltsverzeichnis

1	Diverse kleine Funktionen	1
2	Die totale Quersumme	2
3	Primzahlen	2

1 Diverse kleine Funktionen

1. Ziel:

Der Lernende kann aus der Aufgabenbeschreibung die Funktionen erstellen und Daten korrekt von/zu Funktionen übergeben.

2. Aufgaben:

a) GetMaxValue

Von zwei übergebenen Zahlen soll die grössere als Returnwert zurückgegeben werden.

b) GetZiffer

Eine Funktion soll eine einzelne Ziffer einer übergebenen ganzen Zahl zurückgeben. Die gewünschte Ziffer wird als „Stelle“ (ab 0) mitgegeben.

z.B. Eingabe : Zahl = 70392, Stelle = 1

Ergebnis: Ziffer = 9

2 Die totale Quersumme

1. Ziel:

Der Lernende kann einen Algorithmus entwerfen und als ANSI-C Funktion umsetzen.

2. Problemstellung:

Die (einfache) Quersumme einer Zahl ist die Summe der einzelnen Ziffern:

z.B. die Quersumme von 2571 lautet: $QSumme = 2+5+7+1 = 15$

Von der totalen Quersumme spricht man, wenn man immer wieder die Quersumme bildet, bis das Resultat nur noch aus einer Ziffer besteht.

z.B. die totale Quersumme von 2571 lautet dann: $TQSumme = 1+5 = 6$

3. Aufgaben:

- 1) Es ist eine Funktion zu schreiben, welche von einer Ganzzahl die einfache Quersumme berechnet und als Returnwert zurück liefert. Die Ziffern sind mit mathematischen Operationen zu separieren. Programmieren und testen Sie die Funktion.

Bsp.

`Quersumme = calcQuersumme(12345);`

- 2) Erstellen Sie eine Funktion zur Berechnung der totalen Quersumme aus einer Ganzzahl.

3 Primzahlen

1. Ziel:

Der Lernende kann aus der Aufgabenbeschreibung den Algorithmus finden und daraus die komplette Funktion erstellen.

2. Problemstellung:

Für die verschiedensten Bereiche wie Faktorzerlegung etc. werden Primzahlen gebraucht. Um Primzahl zu finden, sie aufzulisten oder zu testen gibt es verschiedene Algorithmen. Das einfachste, aber sicherlich auch nicht das schnellste Verfahren ist einfach auszuprobieren, ob die zu testende Zahl durch eine andere teilbar ist.

Eine Theorie besagt, dass es reicht bis zum Teiler $= \sqrt{Zahl}$ zu testen, da grössere Teiler die bereits geprüften Teiler enthalten.

3. Aufgaben:

- 1) Erstellen Sie eine Funktion `istPrimzahl(zahl)`, welche überprüft, ob die übergebene Zahl eine Primzahl ist. Ist es eine Primzahl, so soll TRUE (1) ansonsten FALSE (0) zurück gegeben werden. Prüfen Sie Ihren Algorithmus auch spezielle für die Grenzfälle 0, 1 und 2!
- 2) Es ist eine Funktion `nextPrimzahl(zahl)` zu schreiben, welche die nächst grössere Primzahl zur angegebenen Zahl zurück gibt.
- 3) Es ist eine Funktion `printPrimfaktoren(zahl)` zu schreiben, welche die übergebene Zahl in ihre Primfaktoren zerlegt und auf dem Bildschirm ausgibt.

4 Eigene Bibliothek (DLL) erstellen und einbinden (Fakultativ)

1. Ziel:

Der Lernende kann eigene Funktionen als eine Bibliothek (DLL: Dynamic Link Library) erstellen und in einer Anwendung die Bibliothek einbinden.

2. Problemstellung:

Nach dem Prinzip der Modularisierung besteht eine Anwendung aus wiederverwendbaren Modulen. Eigene Funktionen stellen eine Lösung zu einer bestimmten Problemstellung dar. Die Auslagerung solcher Funktionen in einer Bibliothek stellt die Funktionen als ein Modul Client-Anwendungen zur Verfügung. Eine DLL wird erst zur Laufzeit eingebunden. Eine DLL kann von mehreren Programmen gleichzeitig verwendet werden.

3. Aufgaben:

- 1) Erstellen Sie eine DLL für die in der Aufgabe 1 implementierten Funktionen.
- 2) Erstellen Sie eine Konsole-Applikation, die die Funktionen in der DLL aufruft und testet.

Hinweis: Lösen Sie die Aufgaben nach folgender Anleitung

<https://docs.microsoft.com/en-us/cpp/build/walkthrough-creating-and-using-a-dynamic-link-library-cpp?view=msvc-160>

Historie

Dokument erstellt

X.Cheng

10.08.2021