

Dortmund, den 10. Dezember 2010

Übungen zur Vorlesung EINI (WS 2010)

Blatt 8

Block blau

Es können 4 Punkte erreicht werden.

Abgabedatum: 2. Januar 23:59 Uhr

Hinweise

- 1) Bitte beachten Sie die aktuellen Hinweise unter
<http://www4.cs.uni-dortmund.de/home/maeter/UeEiniET10W/>
Für die Abgabe sind die Dateien: *Aufgabe_08_0.txt*, *Aufgabe_08_1a.cpp*, *Aufgabe_08_1b.cpp* und *Aufgabe_08_1c.cpp* zu erstellen.
- 2) Es ist ratsam, die Programme vor der Abgabe zu kompilieren und auszuführen.
- 3) Verwenden Sie für die Textaufgaben reine Texteditoren. Das Abgabesystem erkennt **keine** Word- oder PDF-Dateien!!!
- 4) In der optionalen Datei *Anmerkungen.txt* können Sie allgemeine Anmerkungen bezüglich Ihrer Lösungen notieren.
- 5) Machen Sie sich im Vorhinein Gedanken, wie Sie die zu erstellenden Programme zu implementieren sind. Machen Sie sich also zuerst einen Programmentwurf.
- 6) Kommentieren Sie Ihre Quelltexte. Das gehört zur Aufgabenstellung und wird mitbewertet.
- 7) Verwenden Sie bei Ihrer Implementierung den C++-Standard (insbesondere bei der Deklaration von Feldern).

Der Wurm (Rekursion)

Es stand sehr schlimm um des Bandwurms Befinden.
Ihn juckte immer etwas hinten.
Dann konstatierte der Doktor Schmidt,
Nachdem er den Leib ihm aufgeschnitten,
Daß dieser Wurm an Würmern litt,
Die wiederum an Würmern litten -

Joachim Ringelnatz 1883-1934

Aufgaben

Aufgabe 0: Grundlagen (1 Punkt)

„Um Rekursion zu verstehen, muss man zunächst Rekursion verstehen“.

Legen Sie für Ihre Antworten eine Text-Datei *Aufgabe_08_0.txt* an.

- Welches Arbeitsprinzip wird bei einer Rekursion eingesetzt (0,2 Punkte) ?
- Was versteht man unter einer Rekursionsverankerung (0,2 Punkte) ?
- Nennen Sie 2 Vorteile von rekursiven Funktionen (0,2 Punkte) .
- Was kann in einer rekursiven Funktion passieren, wenn die Rekursionsverankerung fehlt oder falsch ist (0,2 Punkte) ?
- Warum können iterative Lösungen effizienter als rekursive sein (0,2 Punkte) ?

Speichern Sie Ihre Ergebnisse in der Ergebnisdatei *Aufgabe_08_0.txt*.

„Rekursiv geht meistens schief“!

Aufgabe 1: Summe (3 Punkte)

- Legen Sie eine `cpp`-Datei *Aufgabe_08_1a.cpp* an und implementieren Sie eine *rekursive* Funktion

`unsigned int summe(unsigned int n),`

welche die Summe

$$\sum_{i=0}^n i$$

berechnet. Testen und kommentieren Sie die Funktion mit folgender `main`-Funktion (1 Punkt):

```
1 int main() {
2     cout << "summe (0)      = " << summe(0)    << endl;
3     cout << "summe (15)     = " << summe(15)   << endl;
4     cout << "summe (35)     = " << summe(35)   << endl;
5     cout << "summe (101)    = " << summe(101)  << endl;
6     return 0;
7 }
```

Kopieren Sie das Ergebnis als Block-Kommentar an das Ende der Datei *Aufgabe_08_1a.cpp*.

- Legen Sie eine `cpp`-Datei *Aufgabe_08_1b.cpp* an und implementieren Sie eine *rekursive* Funktion

`unsigned int quersumme(unsigned int x),`

welche die Quersumme einer nicht-negativen Zahl berechnet. Die Quersumme ist definiert als die Summe der Ziffern einer Zahl. So ist zum Beispiel die Quersumme von $372 = 3 + 7 + 2 = 12$. Kommentieren und testen Sie die Funktion mit folgender main-Funktion (1 Punkt):

```

1  int main() {
2      cout << "quersumme(479) = " << quersumme(479) << endl;
3      return 0;
4  }
```

Kopieren Sie das Ergebnis als Block-Kommentar an das Ende der Datei *Aufgabe_08_1b.cpp*.

- c) In Aufgabe 1b haben wir das Zehnersystem zugrunde gelegt. Man kann die Quersumme jedoch auch auf andere Zahlensysteme beziehen. Zum Beispiel hat 372 die Binärdarstellung 101110100; daher ist die „Quersumme zur Basis zwei“ $1+0+1+1+1+0+1+0+0 = 5$. Erweitern Sie die Funktion aus Aufgabe 1b so, dass sie einen zusätzlichen Parameter n erwartet und die „Quersumme zur Basis $n \leq 10$ “ errechnet. Falls $n \leq 1$ ist, soll eine entsprechende Fehlermeldung „Die Basis muss größer als 1 sein ...“ ausgegeben werden. Legen Sie eine cpp-Datei *Aufgabe_08_1c.cpp* an, implementieren, kommentieren und testen Sie die Funktion mit folgender main-Funktion (1 Punkt):

```

1  int main() {
2
3      cout << "quersumme(479)      = ";
4      cout << quersumme(479)      << endl;
5      cout << "quersumme_n(479,10) = ";
6      cout << quersumme_n(479,10) << endl;
7      cout << "quersumme_n(479,2)  = ";
8      cout << quersumme_n(479,2)  << endl;
9      cout << "quersumme_n(479,1)  = ";
10     cout << quersumme_n(479,1)  << endl;
11     return 0;
12 }
```

Kopieren Sie das Ergebnis als Block-Kommentar an das Ende der Datei *Aufgabe_08_1c.cpp*.

Tipp: Setzen Sie den Modulo-Operator % ein.

(Präsenz-) Aufgabe 2: Mathematische Funktion (0 Punkte)

Hinweis: Bei dieser Aufgabe handelt es sich um eine leicht modifizierte Prüfungsaufgabe aus dem Jahr 2008.

Gegeben ist die mathematische Funktion $f : \mathbb{N} \rightarrow \mathbb{R}$ mit der Funktionsvorschrift

$$f(x) = \begin{cases} 1 & \text{falls } x=1 \\ f(x-1) + (1/x) & \text{falls } x \text{ gerade} \\ f(x-1) * x & \text{sonst} \end{cases} \quad (1)$$

Diese soll in eine in eine rekursive C++ Funktion überführt werden.

- Implementieren Sie die Funktion $f(x)$.

```
1  #include <iostream>
2  using namespace std;
3
4
5  //*****
6  //   hier die Funktion einfüegen
7  //*****
8
9  int main (){
10     for (unsigned int i = 0; i < 10; i++)
11         cout << "f(" << i << ") = " << f(i) << endl;
12     return 0;
13 }
```

- Testen Sie die Funktion $f(x)$ mit dem main-Programm und notieren Sie sich die Ergebnisse.