



Nagl, Ranger, Würzberger

Übungen zur Vorlesung „Grundgebiete der Informatik 2: Algorithmen und Programmiertechniken“

— Blatt 2 —

4. Aufgabe Operatoren:

(9 Punkte)

In der Vorlesung wurden einige ganzzahlige Operatoren vorgestellt, deren Verwendung hier geübt werden soll.

- (a) Gegeben sei folgendes Programmfragment. Welche Werte nehmen *a*, *b* und *c* jeweils an?

```
int          a = -3;
unsigned int b =  4;
signed char  c =  0;

a++;          b >>= 1;      c = c - 10;
a <<= 2;      b += 9;       c *= -1;
a  = a / 12;  b  = b % 7;   c += 118;
a |= 42;      b ^= 46;      c = ~c & 170;
```

Beachten Sie, daß in C++ Wertebereichsüberschreitungen nicht erkannt werden und nicht zu Laufzeitfehlern führen. (6 Punkte)

- (b) Geben Sie 6 verschiedene Möglichkeiten an, mit jeweils einem Operator den Wert der Variablen `int a=28;` in 12 zu ändern. Verwenden Sie für jede Möglichkeit einen anderen Operator. (3 Punkte)

5. Aufgabe Verstehen und Kommentieren von C++-Quelltext:

(12 Punkte)

In dieser Aufgabe geht es darum, ein unkommentiertes C++-Programm zu verstehen und zu kommentieren. Nicht alle Konstrukte, die in dem Programm vorkommen, sind schon in der Vorlesung erwähnt worden, denn das Programm soll auch vermitteln, daß man bereits durch aufmerksames Lesen einen Eindruck von der Funktion eines Programms bekommen kann.

- (a) Versuchen Sie, durch aufmerksames Lesen des Quelltexts, das unten angegebene C++-Programm zu verstehen und fügen Sie dann Kommentare ein, die Ihnen beim Verständnis des Programms geholfen hätten. Was berechnet das Programm? (6 Punkte)
- (b) Was hat Ihnen an dem Programm gefallen? Was nicht? Geben Sie ggf. kurz Verbesserungsvorschläge an. (4 Punkte)
- (c) Angenommen, in der Eingabe kämen die Zeichen *a* und *b* gleich häufig vor und alle anderen Zeichen wären seltener. Würde das Programm *a* oder *b* ausgeben? Begründen Sie Ihre Antwort kurz. (2 Punkte)

Hinweise

- Das Ende der Eingabe lässt sich via Tastatur mit `Control-Z` (Windows) erzeugen (`Control-D` unter Linux). Eventuell ist vorher und nachher noch ein Return nötig.

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int main()
{
    unsigned int const ANZAHL_ZEICHEN = 256;
    unsigned int gezaehlteZeichen[ANZAHL_ZEICHEN];
    char gelesenesZeichen;

    for(unsigned int zeichenIndex = 0;
        zeichenIndex < ANZAHL_ZEICHEN;
        ++zeichenIndex) {
        gezaehlteZeichen[zeichenIndex] = 0;
    }

    while (cin.get(gelesenesZeichen)) {
        ++gezaehlteZeichen[gelesenesZeichen];
    }

    unsigned int haeufigstesZeichen = 0;
    unsigned int maximaleHaeufigkeit = 0;

    for (unsigned int zeichenIndex = 0;
        zeichenIndex < ANZAHL_ZEICHEN;
        ++zeichenIndex) {
        if (gezaehlteZeichen[zeichenIndex] > maximaleHaeufigkeit) {
            haeufigstesZeichen = zeichenIndex;
            maximaleHaeufigkeit = gezaehlteZeichen[zeichenIndex];
        }
    }
    cout << "Ein haeufigstes Zeichen ist '" << (char)haeufigstesZeichen
        << "' mit " << maximaleHaeufigkeit << " Vorkommen." << endl;
    return 0;
}
```

Zusatzaufgabe Überlegen Sie sich, auf welche Randbedingungen sich das Programm verläßt. Welche Konstrukte sind zwar in C++ üblich, aber nicht sauber?