



Nagl, Ranger, Wörzberger

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

— Blatt 4 —

8. Aufgabe *Histogramm-Programm strukturieren:*

(10 Punkte)

Das Histogramm-Programm hat einen Umfang erreicht, bei dem eine weitere Strukturierung mit Unterprogrammen sinnvoll ist. Dem Hauptprogramm sollte beispielsweise direkt anzusehen sein, welche Schritte unternommen werden.

- (a) Strukturieren Sie das erweiterte Histogramm-Programm aus Aufgabe 7¹ mit Unterprogrammen! Achten Sie darauf, sinnvolle Namen für die Funktionen zu finden. Benutzen Sie keine globalen Variablen. Überlegen Sie, wie sie die Parameter übergeben. Beachten Sie dabei auch die folgenden Fragen. *(8 Punkte)*
- (b) Wofür wird hier noch das häufigste Zeichen ermittelt? Sollte das Hauptprogramm überhaupt etwas davon ‘wissen’? *(1 Punkte)*
- (c) Bei der Ausgabe stört, daß sie ‘nebenbei’ Buchstaben und Whitespace-Zeichen zählt. Beheben Sie das, indem sie die Berechnung in eine separate Funktion auslagern. Benutzen sie keine Referenzen, sondern Zeiger als Rückgabemechanismus. Welche Vorteile bietet dies gegenüber Referenzen? *(1 Punkte)*

¹Die Lösung ist hier abgedruckt und auch in der Nähe der Vorlesungs-Homepage zu haben.

Implementation histogramm.cc

```
#include <math.h>      // Fuer log10() (Logarithmus)
#include <iostream>
#include <iomanip>     // Fuer setw() (ostream-Feldbreite einstellen)
#include <ctype.h>    // Fuer isprint() etc. (character types)

using namespace std;

int main() {
    unsigned int const ANZAHL_ZEICHEN = 256; // Die Anzahl der unterschiedlichen Zeichen.
    unsigned int gezaehlteZeichen[ANZAHL_ZEICHEN]; // Fuer jedes Zeichen ein Zaehler
    // Zusammenfassend:
    unsigned int anzahlBuchstaben = 0; // Buchstaben
    unsigned int anzahlWhitespace = 0; // Leerzeichen etc.
    char gelesenesZeichen; // Das zuletzt gelesene Zeichen
    unsigned int maximaleHaeufigkeit = 0;

    for(unsigned int zeichenIndex = 0 // Initialisiere alle Zaehler mit 0.
        ; zeichenIndex < ANZAHL_ZEICHEN
        ; ++zeichenIndex) {
        gezaehlteZeichen[zeichenIndex] = 0;
    }

    cin.get(gelesenesZeichen); // Lese die Eingabe und erhoehe die entsprechenden Zaehler.
    while (!cin.eof()) {
        ++gezaehlteZeichen[(unsigned int)gelesenesZeichen];
        cin.get(gelesenesZeichen);
    }

    /* Breite der Zahlenausgabe fuer eine huebschere Tabelle bestimmen.
       Dabei beachten, dass log10(0) undefiniert ist. */
    for (unsigned int zeichenIndex = 0
        ; zeichenIndex < ANZAHL_ZEICHEN
        ; ++zeichenIndex) {
        if (gezaehlteZeichen[zeichenIndex] > maximaleHaeufigkeit) {
            maximaleHaeufigkeit = gezaehlteZeichen[zeichenIndex];
        }
    }
    int const BREITE_ANZAHL = (int)(log10(double((maximaleHaeufigkeit>0 ? maximaleHaeufigkeit:1 ))))+1;
    int const BREITE_CODE = (int)(log10(double(ANZAHL_ZEICHEN>0 ? ANZAHL_ZEICHEN:1 )))+1;

    cout << "Die gelesenen Zeichen und ihre Anzahlen:" << endl;
    for(unsigned int zeichenIndex = 0 // Fuer jedes Zeichen den Zaehler geeignet ausgeben.
        ; zeichenIndex < ANZAHL_ZEICHEN
        ; ++zeichenIndex) {
        if (0 != gezaehlteZeichen[zeichenIndex]) {
            cout << "`";
            if ( isprint((char)zeichenIndex) ) {
                cout << (char)zeichenIndex;
            } else {
                cout << "<undruckbares Zeichen, Code "
                    << setw(BREITE_CODE) << zeichenIndex << ">";
            }
            cout << ": " << setw(BREITE_ANZAHL)
                << gezaehlteZeichen[zeichenIndex] << endl;

            if ( isalpha((char)zeichenIndex) ) {
                anzahlBuchstaben += gezaehlteZeichen[zeichenIndex];
            }
            if ( isspace((char)zeichenIndex) ) {
                anzahlWhitespace += gezaehlteZeichen[zeichenIndex];
            }
        }
    }
    cout << "Anzahl Buchstaben: " << anzahlBuchstaben << endl;
    cout << "Anzahl Whitespace: " << anzahlWhitespace << endl;
    return 0;
}
```

9. Aufgabe Zeiger & andere Fallen:

(10 Punkte)

(a) Was geben die folgenden Programme aus?

1.	2.	3.
<pre>#include <iostream> using namespace std; void fool(short a, short *b, short &c) { *b = a; c = *b; } int main() { short a, b, c; a = 1; b = 2; c = 3; fool(a, &b, c); cout << a << ", " << b << ", " << c << endl; }</pre>	<pre>#include <iostream> using namespace std; void foo2(short a, short *b, short &c) { a = c; *b = a; } int main() { short a, b, c; a = 1; b = 2; c = 3; foo2(a, &b, c); cout << a << ", " << b << ", " << c << endl; }</pre>	<pre>#include <iostream> using namespace std; void foo3(short a, short *b, short &c) { a = b[0]; *b = c; } int main() { short a, b, c; a = 1; b = 2; c = 3; foo3(a, &b, c); cout << a << ", " << b << ", " << c << endl; }</pre>

Diese Aufgabe kann, muß aber nicht am Rechner gelöst werden.

(3 Punkte)

(b) In C++ ist die Gefahr relativ groß, daß einfache Schreibfehler zu Programmen führen, die der Compiler zwar ohne Warnung/Fehlermeldung übersetzt, die aber nicht das machen was der Autor erwartet. In den folgenden Programmfragmenten stecken einige typische Fehler. Welche Fehler finden Sie und wie würden Sie sie beheben (ein Satz pro Programm reicht als Erklärung)?

(7 Punkte)

1. <pre>unsigned int werte[4]; unsigned int max=0; for (int x=0; x<4; x++){ werte[x] = x*x - 2*x; if (werte[x]>max) max=werte[x]; } cout <<"Maximum: " <<max;</pre>	2. <pre>char kopie[10]; char original[] ="ein String"; char *o=original; char *k=kopie; while (*o) *k++=*o++; cout << "Kopie:" <<kopie;</pre>	3. <pre>int max=10; for (int i=0;; i++) { cout << i << ", "; if (i==max) break; } cout << "schon Fertig!";</pre>
4. <pre>int i:=0; while (i<10) cout << i << ", "; i++; cout << "Fertig!";</pre>	5. <pre>int a=3, b=4; if (a & b) cout <<"beide ungleich 0";</pre>	6. <pre>int a=0, b=0; if (! a==0 b==0) cout <<"beide ungleich 0";</pre>

