Lehrstuhl für Informatik 3
Rheinisch-Westfälische Technische Hochschule
Aachen
Univ.-Prof. Dr.-Ing. M. Nagl

Aachen, 27. 04. 2007

Nagl, Ranger, Wörzberger

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

— Blatt 3 —

6. Aufgabe *Kontrollstrukturen*: *(10 Punkte)*

Hier soll der Umgang mit verschiedenen Kontrollstrukturen eingeübt werden. Gegeben ist jeweils ein Programmfragment, das so umgestellt werden soll, daß statt der gegebenen Kontrollstruktur eine andere verwendet wird. Alle Fragmente verwenden folgenden Programmrahmen:

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

using namespace std;

int main()
{
  int i, j;       char c;
  /* ... Programmfragment ... */
  return 0;
}
```

(a) Bedingte Anweisungen und Auswahlanweisungen. *(4 Punkte)*

1. Ersetzen Sie das `switch`-Statement durch entsprechende `if`-Statements.

```
cout << "Was tun?";    cin >> c;
switch(c) {
  case 'n': cout << "Neue Datei anlegen" << endl; break;
  case 'u': cout << "Datei umbenennen"   << endl; break;
  case 'l': cout << "Datei loeschen"     << endl; break;
  default : cout << "Unbekannte Eingabe" << endl;
}
```

2. Ersetzen Sie hier umgekehrt die `if`-Statements durch *eine* `switch`-Anweisung. Was fällt auf?

```
cout << "Zahl eingeben (0..15)";  cin >> i;
if ( 0 <= i && i <=  9) {
   cout << "Hex: " << char(i   +'0') << endl;
} else if (10 <= i && i <= 15) {
   cout << "Hex: " << char(i-10+'A') << endl;
} else {
   cout << "Zahl nicht im legalen Bereich" << endl;
}
```

(b) Kontrollierte Sprünge. *(4 Punkte)*

1. Durch welchen kontrollierten Sprung sollte hier das `goto` ersetzt werden?

```
cout << "Zahl eingeben:";   cin >> i;
j = 2;
while (j*j <= i) {
   if (0 == i%j) { goto fertig; }
   j++;
}
fertig:
if (j*j <= i) { cout << "keine Primzahl" << endl; }
else          { cout <<       "Primzahl" << endl; }
```

2. Geben Sie ein äquivalentes Programmfragment an, das ohne `continue` (und auch ohne `goto`) auskommt. Welche Variante gefällt Ihnen besser?

```
i = 0;   c=' ';
while (c != '.' && NULL != cin.get(c)) {
   if ('\n' != c) { continue; } // Return eingegeben?
   i++;
   cout << i << ":";            // Zeilennummer ausgeben
}
```

(c) In C++ kann man mit Hilfe von `for` beliebige Schleifen hinschreiben. Eigentlich sollte man `for` aber nur für Zählschleifen verwenden, bei denen die Anzahl der Schleifendurchläufe festliegt. Ersetzen Sie deshalb folgende `for`-Schleife durch eine äquivalente `while`-Schleife. *(2 Punkte)*

```
char text[] = "Wie lang ist der Text?";
for( i = 0;  0 != text[i]; ++i ) { /* Nixtun */ };
cout << "Der Text ist " << i << " Zeichen lang" << endl;
```

7. Aufgabe *Histogramm erweitern*: *(9 Punkte)*

Ersetzen Sie die ungenaue Ausgabe des Histogramm-Programms aus Aufgabe 5 durch eine aussagekräftigere Ausgabe! Für jedes Zeichen, das mindestens einmal vorkommt, soll es die Anzahl angeben; zusammenfassend soll es die Anzahl Buchstaben und die Anzahl der Weißraumzeichen (Leerzeichen, Tabulatoren etc.) ausgeben.

Benutzen Sie dazu aus Header `ctype.h` (in ISO C++: `limits.h`) die Funktionen `isprint`, `isalpha` und `isspace`, die je ein Zeichen als Argument nehmen und `true` liefern, falls es druckbar, ein Buchstabe oder Weißraum ist.

Hinweis: Das Programm liest von der *Standard-Eingabe*, bis diese *leer* ist. Wird das compilierte Programm 'einfach so' gestartet, ist die Standard-Eingabe die Tastatur. Das Ende der Eingabe teilt man mit *Control-D*, unter DOS-Derivaten und Windows NT *Strg-Z* mit. Auf UN∗X(-ähnlichen) Systemen kann man die Standard-Eingabe aus einer Datei füttern, etwa mit `./histogramm < histogramm.cc`.

Abgabe: 04. 05. 2007 vor der Vorlesung.