Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für theoretische Informatik
Dr. B. Gärtner

30. November 2004

Informatik für Mathematiker und Physiker Serie 7 WS 04/05

URL: `http://www.ti.inf.ethz.ch/ew/courses/Info1_04/`

Allgemeine Bemerkungen: Wann immer Sie auf dieser oder einer der folgenden Serien eine Funktion schreiben sollen, gehört dazu die Angabe von Pre– und Postcondition(s). Ausserdem erstellen Sie ein Hauptprogramm, das die jeweilige Funktion testet. Sowohl die Funktion als auch das Testprogramm gehören zur Abgabe.

Aufgabe 1 [Schnellübung – 20 Min.] (5 Punkte)

Gruppe A–H

Eine natürliche Zahl n heisst *vollkommen* genau dann, wenn sie die Summe ihrer echten Teiler ist. (Echte Teiler sind alle Teiler ausser der Zahl selbst.) Zum Beispiel ist

$$28 = 1 + 2 + 4 + 7 + 14$$

vollkommen.

Schreiben Sie eine C++-Funktion, die prüft, ob eine gegebene Zahl n vollkommen ist.

Gruppe I

Schreiben Sie eine C++-Funktion, die prüft, ob für eine gegebene Zahl n die Anzahl ihrer Teiler gerade ist. Zum Beispiel ist

$$\{1, 2, 4, 7, 14, 28\}$$

die Menge der Teiler von 28; also lautet die Antwort in diesem Fall "JA".

Gruppe J und K

Schreiben Sie eine C++-Funktion, die prüft, ob für eine gegebene Zahl n die Summe ihrer Teiler gerade ist. Zum Beispiel ist

$$\{1, 2, 4, 7, 14, 28\}$$

die Menge der Teiler von 28; da 54 gerade ist, lautet die Antwort in diesem Fall also "JA".

Aufgabe 2 (4 Punkte)

Schreiben Sie eine Funktion

```
unsigned int binomial(unsigned int n, unsigned int k)
```

die den Binomialkoeffizienten $\binom{n}{k}$ rekursiv berechnet. Führen Sie dabei alle Berechnungen auf dem Zahlentyp `unsigned int` durch.

Testen Sie Ihr Programm, insbesondere mit $\binom{30}{15} = 155117520$.

Aufgabe 3 (4 Punkte)

Geben Sie reguläre Ausdrücke für folgende Sprachen über dem Alphabet $\Sigma = \{0, 1\}$ an.

a) Alle Wörter, die 1001 als Teilwort enthalten.

b) Alle Wörter der Länge $3i$, für $i \geq 0$.

c) Alle Wörter mit einer geraden Anzahl von Einsen.

d) Alle Wörter, worin alle Teilwörter der Länge vier mindestens eine 1 enthalten.

Aufgabe 4 (4 Punkte)

Schreiben Sie eine Funktion

```
unsigned int wordcount(const std::string& s)
```

welche die Anzahl von Wörtern in `s` zurückgibt. Als Wörter zählen hierbei maximale nichtleere Folgen von nicht-whitespace-Zeichen. Sie können die in der Bibliothek `<cctype>` definierte Funktion `bool std::isspace(char c)` verwenden, die `true` zurückgibt genau dann, wenn `c` ein whitespace-Zeichen ist.

Testen Sie Ihr Programm, insbesondere besteht der Text `freude.txt` auf der Vorlesungswebseite aus 377 Wörtern.

Abgabe: Aufgabe 2+4: bis 6. Dezember 2004, 16.00 Uhr, per Email.
Aufgabe 3: am 7. Dezember 2004, in der Pause der Vorlesung, schriftlich.

Eidgenössische Technische Hochschule Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für theoretische Informatik
Dr. B. Gärtner

30. November 2004

Informatik I: Material aus der Vorlesung

Programm: mac2unix-index.C

```
// Programm: max2unix-index.C
// Konvertiert MAC textfiles in UNIX textfiles

#include <iostream>
#include <string>

void mac_to_unix(std::string& s)
// POST: In s wurde jedes Auftreten von linefeed (LF)
//   durch carriage-return (CR) ersetzt.
{
  for (unsigned int i = 0; i < s.length(); ++i)
    if (s[i] == '\r') s[i] = '\n';
}

int main()
{
  std::string f;
  // Lies eine ganze Zeile auf einmal ein.
  std::getline(std::cin, f);

  mac_to_unix(f);
  std::cout << f;
  return 0;
}
```

Programm: mac2unix.C

```
// Programm: max2unix.C
// Konvertiert MAC textfiles in UNIX textfiles

#include <iostream>
#include <string>

void mac_to_unix(std::string& s)
// POST: In s wurde jedes Auftreten von linefeed (LF)
//   durch carriage-return (CR) ersetzt.
{
  typedef std::string::iterator Sit;
  for (Sit i = s.begin(); i != s.end(); ++i)
    if (*i == '\r') *i = '\n';
}

int main()
{
  std::string f;
  // Lies eine ganze Zeile auf einmal ein.
  std::getline(std::cin, f);

  mac_to_unix(f);
  std::cout << f;
  return 0;
}
```

Programm: mac2unix-iterator.C

```
// Programm: max2unix-iterator.C
// Konvertiert MAC textfiles in UNIX textfiles

#include <iostream>
#include <string>

void mac_to_unix(std::string::iterator b,
                 std::string::iterator e)
// POST: Im range [b,e) wurde jedes Auftreten von
//   linefeed (LF) durch carriage-return (CR) ersetzt.
{
  for (; b != e; ++b)
    if (*b == '\r') *b = '\n';
}

int main()
{
  std::string f;
  // Lies eine ganze Zeile auf einmal ein.
  std::getline(std::cin, f);

  mac_to_unix(f.begin(), f.end());
  std::cout << f;
  return 0;
}
```

Programm: linecount.C

```
// Programm: linecount.C
// Zaehlt Zeilen in einem String

#include <iostream>
#include <string>

unsigned int linecount(const std::string& s)
// POST: Rueckgabewert ist Anzahl der Zeilen in s.
{
  unsigned int c = 0;
  typedef std::string::const_iterator Scit;
  for (Scit i = s.begin(); i != s.end(); ++i)
    if (*i == '\n') ++c;
  return c;
}

int main()
{
  // Lies eine ganze Datei auf einmal ein
  //   --> bis eof = end of file
  std::string f;
  typedef std::istream::traits_type CT;
  std::getline(std::cin, f, CT::to_char_type(CT::eof()));

  std::cout << "Die Datei besteht aus " << linecount(f)
            << " Zeile(n)." << std::endl;
  return 0;
}
```