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

Institut für theoretische Informatik
Dr. J. Giesen

20. Oktober 2004

Informatik I für D-ITET — Serie 1 — WS 04/05

URL: `http://www.ti.inf.ethz.ch/ew/courses/inf1-ITET/`

Die erste Übung soll ein wenig mit der Arbeitsumgebung auf einer Unix Workstation vertraut machen. Wer auf einem eigenen Rechner arbeitet, sollte ggf. benötigte Software einrichten; eine Kurzanleitung zur Installation von cygwin sowie eine Liste wichtiger Unix Kommandos findet man auf der Vorlesungswebseite. Ab der kommenden Woche wird eine funktionierende Arbeitsumgebung vorausgesetzt.

Aufgabe 1 (4 Punkte)

Kopieren Sie die Datei `.emacs` von der Vorlesungswebseite in Ihr Heimatverzeichnis. Legen Sie in Ihrem Heimatverzeichnis ein Unterverzeichnis `ifm` an. Kopieren Sie dorthin die Datei `prime.C` von der Vorlesungswebseite. Wechseln Sie ins Verzeichnis `~/ifm` und starten Sie den `emacs`.

Öffnen Sie die Datei `prime.C` und übersetzen Sie sie. Starten Sie es im shell-Terminal mit `./prime` .

Im Programmtext sind drei Stellen markiert, an denen absichtlich ein Fehler eingebaut werden soll. Probieren Sie die Fehler einzeln nacheinander aus und beachten Sie die Fehlermeldung beim Compilieren. Probieren Sie auch aus, wie sich die automatische Einrückung jeweils verändert.

Tip: Folgende `emacs`-Befehle sind hierbei hilfreich.

[F9]	übersetze den Quelltext in ein ausführbares Programm.
[F8]	gehe zum nächsten Fehler im Quelltext.
[F7]	gehe zum vorhergehenden Fehler im Quelltext.
[F6]	rücke den gesamten Quelltext automatisch ein.
[TAB]	rücke die aktuelle Zeile im Quelltext ein.

Aufgabe 2 (4 Punkte)

Ändern Sie die Datei `prime.C`, so dass sie den Anforderungen genügt, die wir an Übungsabgaben stellen (siehe das Merkblatt zur Vorlesung). Insbesondere muss am Anfang der Datei ein Kommentar der Form

```
// Informatik - Serie 1 - Aufgabe 2
// Programm: prime.C
// Autor:    X. M. Plestudent (Gruppe D)
```

stehen.

Schicken Sie eine Email mit der Datei `prime.C` (in der korrekten Fassung) als *attachment* an Ihren Übungsleiter.

Abgabe: Aufgabe 2: bis Montag 1. November 2004, 14.00 Uhr, per Email.

Eidgenössische Technische Hochschule Zürich

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

Institut für theoretische Informatik
Dr. J. Giesen

20. Oktober 2004

Informatik I: Material aus der Vorlesung

Programm: prog2.C

```
#include <iostream>

int main()
{
  int sum;
  sum = 5 + 7;
  std::cout << sum;
  std::cout << std::endl;
  return 0;
}
```

Programm: prog3.C

```
#include <iostream>

int main()
{
  int x;
  int y;
  int sum;
  std::cin >> x;
  std::cin >> y;
  sum = x + y;
  std::cout << sum;
  std::cout << std::endl;
  return 0;
}
```

Programm: summe.C

```
// Programm: summe.C
// Berechnet die Summe
// zweier Eingabezahlen.

#include <iostream>

int main()
{
  int x;    // 1. Summand
  int y;    // 2. Summand
  int sum;  // Summe

  // Eingabe
  std::cout << "Eingabe x: ";
  std::cin >> x;
  std::cout << "Eingabe y: ";
  std::cin >> y;

  // Berechnung
  sum = x + y;

  // Ausgabe
  std::cout << "Summe von " << x
            << " und " << y << " ist "
            << sum << "." << std::endl;
  return 0;
}
```

Programm: prog1.C

```
int main()
{
  int sum;
  sum = 5 + 7;
  return 0;
}
```

Programm: prime.C

```
// Programm: prime.C
// Testet, ob eine Zahl Primzahl ist.
// Wenn nicht, Ausgabe des kleinsten Primteilers.

#include <iostream>

int main ()
{
  // Eingabezahl
  int Zahl;
  // Potentielle Teiler der Eingabezahl
  int Teiler = 1;
  // Rest bei Division von Zahl durch Teiler
  int Rest;

  // Einlesen der Eingabe
  std::cout << "Zahl? (2 <= Zahl <= 2147483647) ";
  std::cin >> Zahl;

  // Test, ob Teiler Zahl teilt. Abbruch
  // spaetestens bei Teiler == Zahl.
  do {
    Teiler = Teiler + 1;
    Rest = Zahl % Teiler;
  } while (Rest != 0);

  // Ausgabe des Ergebnisses
  if (Teiler == Zahl)
    std::cout << Zahl << " ist Primzahl." << std::endl;
  else
    std::cout << Zahl << " ist keine Primzahl und " << Teiler
              << " ist der kleinste Primteiler von " << Zahl
              << "." << std::endl;
  return 0;
}
```

Programm: absurd.C

```
#include <cstdio>
int v,i,j,k,l,s,a[99];
int main()
{
  for(scanf("%d",&s);*a-s;v=a[j*=v]-a[i],
  k=i<s,j+=(v=j<s&&(!k&&!!printf(2+
  "\n\n%c"-(!l<<!j)," #Q"[l^v?(l^j)
  &1:2])&&++l||a[i]<s&&v&&v-i+j&&v+i-j))
  &&!(l%=s),v||(i==j?a[i+=k]=0:++a[i])>=
  s*k&&++a[--i]);
  return 0;
}
```