**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*

*27. Oktober 2004*

# Informatik I für D-ITET Serie 2 WS 04/05

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

## Aufgabe 1 (4 Punkte)

Werten Sie folgende Ausdrücke „per Hand“ **in Einzelschritten** und in der durch Priorität und Links-/Rechtsassoziativität bestimmten Reihenfolge aus. Zum Beispiel

$$3 + 7 * 5/3 - 3 \longrightarrow 3 + 35/3 - 3 \longrightarrow 3 + 11 - 3 \longrightarrow 14 - 3 \longrightarrow 11\ .$$

a) `6 + 12 % 5 * 2 + 3 * 2`

b) `5 * 4 / 6 * 3`

c) `5 / 6 * 4 * 3`

d) `5 * 4 * 3 / 6`

## Aufgabe 2 (4 Punkte)

Kopieren Sie sich das Programm `binaer.C` von der Vorlesungswebseite. Verändern Sie das Programm so, dass neben der darzustellenden Zahl auch noch eine Basis $b (2 \leq b \leq 10)$ eingegeben werden kann. Die Darstellung der Zahl soll dann zur Basis $b$ erfolgen. Zum Beispiel gilt

$$45 = 1 \cdot 3^3 + 2 \cdot 3^2 + 0 \cdot 3^1 + 0 \cdot 3^0,$$

also soll als (umgekehrte) Darstellung von 45 zur Basis 3 die Folge `0021` ausgegeben werden.

## Aufgabe 3 (4 Punkte)

Als der Mathematiker Carl-Friedrich Gauss noch zur Schule ging, erhielt seine Klasse vom Lehrer einmal eine Aufgabe, mit der er sie länger beschäftigen wollte: die Schüler sollten die Zahlen von 1 bis 100 aufaddieren. Gauss' Mitschüler hatten kaum angefangen zu rechnen, als er bereits die richtige Lösung verkündete: 5050. Dieser schnellen Berechnung liegt die Formel $\sum_{i=1}^{n} i = n(n+1)/2$ zugrunde, die der kleine Gauss damals während der Unterrichtsstunde entdeckte.

Überprüfen Sie diese Formel, indem Sie ein Programm schreiben, das für eine Eingabezahl $n \geq 0$ die Summe einerseits auf traditionelle Weise mit einer `while`-Schleife berechnet, andererseits direkt über den Ausdruck $n(n+1)/2$. Beide Werte sollen ausgegeben werden, und falls sie übereinstimmen, soll die Zeichenkette `‘‘Gauss hatte recht!’’` ausgegeben werden.

**Abgabe:** Aufgabe 1: am 1. November 2004, am Anfang der Übungen, schriftlich.
Aufgabe 2 und 3: bis 8. 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*

*27. Oktober 2004*

# Informatik I: Material aus der Vorlesung

## Programm: binaer.C

```
// Programm: binaer.C
// berechnet die Binaerdarstellung einer Dezimalzahl

#include <iostream>
#include <limits>

int main()
{
  // Eingabe
  std::cout << "Eingabezahl (0 <= zahl <= "
            << std::numeric_limits<unsigned int>::max()
            << ")" << std::endl;
  unsigned int zahl;
  std::cin >> zahl;

  // Berechnung / Ausgabe
  std::cout << "Umgekehrte Binaerdarstellung von "
            << zahl << " ist: ";
  do {
    std::cout << zahl % 2; // gib letzte Binaerstelle aus
    zahl = zahl / 2;       // und entferne sie
  } while (zahl != 0);
  std::cout << "." << std::endl;
  return 0;
}
```

## Programm: zufall.C

```
// Programm: random.C
// berechnet eine Folge von Pseudozufallszahlen

#include <iostream>

int main()
{
  // Lineare Kongruenzmethode zur Erzeugung von
  // Pseudozufallszahlen x_0, x_1, x_2, ...
  // mit Hilfe der Formel x_i = a * x_(i-1) % m

  const unsigned int m = 65536;  // Modulus, 2^16
  const unsigned int a = 47485;  // Multiplikator
  const unsigned int x0 = 1;     // Startwert

  // lies Anzahl zu erzeugender Zahlen ein
  std::cout << "Anzahl von Pseudozufallszahlen: ";
  unsigned int anzahl;
  std::cin >> anzahl;

  // Erzeugung und Ausgabe
  unsigned int x = x0;  // aktuelle Zufallszahl
  std::cout << "Folge von " << anzahl
            << " Zufallszahlen: ";
  while (anzahl > 0) {
    std::cout << x << " ";
    x = a * x % m;
    anzahl = anzahl - 1;
  }
  std::cout << std::endl;
  return 0;
}
```

## Programm: muenzwurf.C

```
// Programm: muenzwurf.C
// berechnet eine Folge von Pseudozufallszahlen,
// simuliert damit Muenzwuerfe und ueberprueft
// deren Verteilung statistisch

#include <iostream>

int main()
{
  // Lineare Kongruenzmethode zur Erzeugung von
  // Pseudozufallszahlen x_0, x_1, x_2, ...
  // mit Hilfe der Formel x_i = a * x_(i-1) % m

  const unsigned int m = 65536;  // Modulus, 2^16
  const unsigned int a = 47485;  // Multiplikator
  const unsigned int x0 = 1;     // Startwert

  // lies Anzahl zu werfender Muenzen ein
  std::cout << "Anzahl von Muenzwuerfen: ";
  unsigned int anzahl;
  std::cin >> anzahl;

  // Erzeugung und Ausgabe
  unsigned int x = x0;  // aktuelle Zufallszahl
  std::cout << "Folge von " << anzahl
            << " Muenzwuerfen: ";
  unsigned int kopf = 0; // # "Kopf"-Wuerfe
  unsigned int zahl = 0; // # "Zahl"-Wuerfe

  while (anzahl > 0) {
    // fuehrendes Bit der Zufallszahl
    unsigned int muenzwurf = x / (m / 2);
    std::cout << muenzwurf << " ";
    if (muenzwurf == 0)
      kopf = kopf + 1;
    else
      zahl = zahl + 1;
    x = a * x % m;
    anzahl = anzahl - 1;
  }
  std::cout << "\nKopf: " << kopf << "-mal"
            << "\nZahl: " << zahl << "-mal"
            << std::endl;
  return 0;
}
```

## Programm: fahrenheit.C

```
// Programm: fahrenheit.C
// rechnet eine Temperatur von Grad Celsius
// nach Grad Fahrenheit um

#include <iostream>

int main()
{
  // Eingabe
  std::cout << "Temperatur in Grad Celsius (ganzzahlig) ? ";
  int celsius;
  std::cin >> celsius;

  // Berechnung / Ausgabe
  int fahrenheit = 9 / 5 * celsius + 32;
  std::cout << celsius << " Grad Celsius entsprechen "
            << fahrenheit << " Grad Fahrenheit,"
            << "\ngemaess der Formel: "
            << "fahrenheit = 9/5*celsius+32"
            << std::endl;
  fahrenheit = 9 * celsius / 5 + 32;
  std::cout << celsius << " Grad Celsius entsprechen "
            << fahrenheit << " Grad Fahrenheit,"
            << "\ngemaess der Formel: "
            << "fahrenheit = 9*celsius/5+32"
            << std::endl;
  return 0;
}
```