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, Prof. Dr. J. Hromkovič*

*27. Oktober 2009*

# Informatik für Mathematiker und Physiker Serie 6 HS 09

URL: http://www.ti.inf.ethz.ch/ew/courses/Info1_09/

## Skript-Aufgabe 59 (4 Punkte)

Compute the binary expansions of the following decimal numbers.

*a)* 0.25 *b)* 1.52 *c)* 1.3 *d)* 11.1

## Skript-Aufgabe 65 (4 Punkte)

What is the problem with the following loop (assuming the IEEE standard 754)?

```
for (float i = 0.0f; i < 100000000.0f; ++i)
   std::cout << i << "\n";
```

## Skript-Aufgabe 68 (4 Punkte)

The number $\pi$ can be defined through various infinite sums. Here are two of them.

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots$$
$$\frac{\pi}{2} = 1 + \frac{1}{3} + \frac{1 \cdot 2}{3 \cdot 5} + \frac{1 \cdot 2 \cdot 3}{3 \cdot 5 \cdot 7} + \cdots$$

Write a program for computing an approximation of $\pi$, based on these formulas. Which formula is better for that purpose?

## Skript-Aufgabe 69 (4 Punkte)

There is a well-known iterative procedure (the *Babylonian method*) for computing the square root of a positive real number $s$. Starting from any value $x_0 > 0$, we compute a sequence $x_0, x_1, x_2, \ldots$ of values according to the formula

$$x_n = \frac{1}{2}(x_{n-1} + \frac{s}{x_{n-1}}).$$

It can be shown that

$$\lim_{n \to \infty} x_n = \sqrt{s}.$$

Write a program `babylonian.cpp` that reads in the number $s$ and computes an approximation of $\sqrt{s}$ using the Babylonian method. To be concrete, the program should output the first number $x_i$ such that

$$|x_i^2 - s| < 0.001.$$

Die **Aufgaben 73**, **74** und **75** aus den Vorlesungsunterlagen sind die **Challenge Aufgaben** und geben jeweils 8 Punkte, wenn sie vollständig gelöst werden.

**Abgabe:** Bis 3. November 2009, 15.15 Uhr.