

Informatik I für D-ITET

Serie 4

WS 04/05

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

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

Schreiben Sie ein Programm `even.C`, das eine natürliche Zahl n einliest und die Summe der ersten n geraden Zahlen berechnet und ausgibt.

Aufgabe 2 (2 Punkte)

Werten Sie folgende Ausdrücke „per Hand“ in Einzelschritten und in der durch Priorität und Links-/Rechtsassoziativität sowie den Regeln für boolesche Ausdrücke bestimmten Reihenfolge aus. Zum Beispiel:

```
3 + 4 < 8 && 3 == 5 - 3
→ 7 < 8 && 3 == 5 - 3
→ true && 3 == 5 - 3
→ true && 3 == 2
→ true && false
→ false.
```

a) `5 * 7 <= 32 && 11 == 2 * 5 + 1 || 6 > 5`

b) `3 * 2 > 2 || 1.0 / 0.0 != 0.0 && 3 + 4 >= 7 || 2 < 8 + 0.876`

Aufgabe 3 (5 Punkte)

Schreiben Sie ein Programm `randmax.C`, das eine natürliche Zahl n einliest und das Maximum von n Zufallszahlen berechnet. Das Programm soll ausgeben, wie oft sich das Maximum im Verlaufe der Berechnung geändert hat. Vergleichen Sie die Ergebnisse für verschiedene n mit denen des Programms `harmonic`.

Hinweis: Verwenden Sie den Pseudozufallszahlengenerator aus der Vorlesung. Das Programm `random.C` finden Sie auf der Vorlesungswebseite.

Abgabe: Aufgabe 2: am 15. November 2004, am Anfang der Übungen, schriftlich.
Aufgabe 3: bis 22. November 2004, 12.00 Uhr, per Email.

Informatik I:

Material aus der Vorlesung

Programm: newton.C

```
// Programm: newton.C
// Berechnet Quadratwurzeln mittels Newton-Iteration.

#include <iostream>

int main()
{
    // Eingabe
    std::cout << "Berechnung der Quadratwurzel von n,"
               << " n > 0. Eingabe n :? ";

    double n;
    std::cin >> n;

    // Newton-Iteration:  $x_{i+1} = 1/2 (x_i + n/x_i)$ 
    // Um sicherzustellen, dass der Algorithmus terminiert,
    // halten wir eine untere Schranke fuer den aktuellen
    // Approximationswert aufrecht. Diese Schranke wird
    // hochgesetzt, wann immer der folgende Wert groesser
    // wird als der aktuelle. Auf diese Weise wird ein
    // etwaiger Zyklus erkannt und abgebrochen.

    double curr = n;           // aktueller Wert
    double prev = 0;          // voriger Wert
    double lb = 0;            // untere Schranke
    unsigned int counter = 0; // Anzahl Iterationen

    do {
        ++counter;
        if (curr > prev)
            lb = prev;
        prev = curr;
        curr = (prev + n / prev) / 2;
    } while (curr != prev && curr > lb);

    // Ausgabe
    std::cout << "Approximation der Quadratwurzel aus "
              << n << " ist: " << curr
              << "\nApproximationsfehler des Quadrats ist "
              << curr * curr - n << "."
              << "\nAnzahl der Iterationen war "
              << counter << "." << std::endl;
}
```