

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

## Aufgabe 1

Frühe Übungen: Programm: perfect.h \_\_\_\_\_

```
bool perfect(unsigned int n)
// PRE: n > 0.
// POST: Gibt true zurueck genau dann, wenn die Zahl
//       n vollkommen ist, d.h. sie ist die Summe ihrer
//       echten Teiler.
{
    unsigned int sum = 0; // Summe echter Teiler
    for (unsigned int i = 1; i < n; ++i)
        if (n % i == 0) sum += i;
    return sum == n;
}
```

Späte Übungen: Programm: even-num.h \_\_\_\_\_

```
bool div_even(unsigned int n)
// PRE: n > 0.
// POST: Gibt true zurueck genau dann, wenn die Anzahl
//       der Teiler von n gerade ist.
{
    unsigned int div = 0; // Anzahl Teiler
    for (unsigned int i = 1; i <= n; ++i)
        if (n % i == 0) ++div;
    return div % 2 == 0;
}
```

## Aufgabe 2

### Programm: binomial.C

---

```
// Programm: binomial.C
// Berechnet Binomialkoeffizienten rekursiv.

#include <iostream>

unsigned int binomial(unsigned int n, unsigned int k)
    // POST: Rueckgabewert ist der Binomialkoeffizient "n tief k".
{
    if (k == 0) return 1;
    if (n == 0) return 0;
    return binomial(n-1, k-1) * n / k;
}

int main()
{
    std::cout << "Berechnung von n tief k: n ?= ";
    unsigned int n;
    std::cin >> n;
    std::cout << "k ?= ";
    unsigned int k;
    std::cin >> k;

    std::cout << n << " tief " << k << " = "
              << binomial(n, k) << std::endl;
    return 0;
}
```

## Aufgabe 3

- a)  $(0|1)*1001(0|1)*$
- b)  $((0|1)(0|1)(0|1))*$
- c)  $0*(10*10)*$
- d)  $(1|01|001|0001)*(\epsilon|0|00|000)$

## Aufgabe 4

### Programm: wordcount.C

---

```
// Programm: woerter.C
// Zaehlt Woerter in einem Text.

#include <iostream>
#include <string>
#include <cctype>

unsigned int wordcount(const std::string& s)
{
    unsigned int c = 0;
```

```

for (std::string::const_iterator i = s.begin(); i != s.end();)
    if (std::isspace(*i))
        ++i;
    else {
        // Hier beginnt ein neues Wort
        ++c;
        // Lese bis zum Ende des Wortes
        while (++i != s.end() && !std::isspace(*i)) {}
    }
return c;
}

int main()
{
    typedef std::istream::traits_type CT;
    std::string f;
    std::getline(std::cin, f, CT::to_char_type(CT::eof()));
    std::cout << "Die Datei besteht aus " << wordcount(f)
        << " Wo(e)rt(ern)." << std::endl;
    return 0;
}

```