



**Aufgabe 2.** Primfaktorzerlegung

```
1 #include "cpp_headers/fcpp.hh"
2
3 int main() {
4     int n = enter_int("Please enter a natural number: ");
5     // search smallest factor, start with smallest possible: 2
6     int k=2;
7     // go until sqrt(n)
8     while (k <= sqrt(n)) {
9         // k is factor of n
10        if (n % k == 0) {
11            // print out the factor k
12            print(k);
13            // reset n to the quotient
14            n = n / k;
15            // restart at k=2
16            k = 2;
17        } else {
18            // if k is not a factor, check next one
19            k++;
20        }
21    }
22    // n is the last prime factor of the original input number
23    print(n);
24 }
25
```

## Primfaktorzerlegung

Die algorithmische Komplexität des Programms für  $n$  Primzahl ist  $\sqrt{n}$ , da die Schleife für Primzahlen bis  $\sqrt{n}$  durchlaufen wird.

**Aufgabe 3.** siehe *taschenrechner.cpp*