

Aufgabenstellung der ersten praktischen Aufgabe¹

Erstellen Sie ein Programm „*prim*“, das nacheinander natürliche Zahlen einliest und für jede eingelesene Zahl eine Primfaktorzerlegung durchführt. Bei der Eingabe von 0 oder einer negativen Zahl soll das Programm beendet werden. Folgende Zeile soll zum Beispiel bei der Eingabe von 12 ausgegeben werden:

12 = 2 * 2 * 3

Pro Ausgabezeile muss also ausgegeben werden:

- die eingegebene Zahl
- das Gleichheitszeichen
- danach die Primfaktorzerlegung (sortiert nach der Größe der Primzahlen, mit der kleinsten beginnend und jeweils verbunden durch das Multiplikationszeichen „*“)

Da die Zahl 1 keine Primzahl ist, soll sie auch nicht bei der Zerlegung auftauchen. Eine Ausnahme gibt es hiervon lediglich bei der Eingabe der Zahl 1, die folgende Ausgabe haben soll:

1 = 1

Der Quelltext „*prim.c*“ soll im Prinzip aus folgenden Funktionen bestehen:

```
long long kTeiler(long long n);  
void prim(long long n);  
int main(void);
```

In der Funktion `main()` müssen die Zahlen, deren Primfaktorzerlegung bestimmt werden soll, in einer Schleife mit Hilfe der Funktion `leseLongLong()` eingelesen werden. Für jede eingelesene Zahl soll die Funktion `prim()` aufgerufen werden!

Mit Hilfe der Funktion `kTeiler()` berechnet die Funktion `prim()` die Primfaktorzerlegung der übergebenen Zahl und gibt sie mit `printf()` aus. Die Funktion `kTeiler()` gibt den kleinsten Teiler der übergebenen Zahl zurück und soll nur in der Funktion `prim()` aufgerufen werden.

Testen Sie Ihr Programm mit den Dateien „*primData2*“ und „*primData3*“ (im Verzeichnis „*ueb5*“):

```
prim < primData2   bzw.   prim < primData3
```

Die Laufzeit sollte deutlich unter 2 s für „*primData2*“ (bzw. 100 s für „*primData3*“) liegen.

¹ die genauen Abgabebedingungen werden noch bekanntgegeben