

## Allgemeines

Siehe Testat 0

### Aufgabe 3: Primfaktorzerlegung

Bereits Euklid konnte beweisen, dass jede natürliche Zahl  $z \in \mathbb{N}$ ,  $z > 1$  sich eindeutig als Produkt einer Menge von Primzahlen darstellen lässt.

Entwickeln oder recherchieren Sie einen Algorithmus zur Primfaktorzerlegung und codieren Sie Ihren Algorithmus in C. So ist bspw. die Primfaktorzerlegung von 6936 das Produkt  $2 * 2 * 2 * 3 * 17 * 17$ .

- Das Programm liest eine Variable ein (genau ein scanf)

---

```
scanf ("%d", ...);
```

---

- Das Programm gibt im Normalfall die Primfaktoren untereinander aufsteigend sortiert linksbündig aus:

---

```
printf ("%d\n", ...);
```

---

(natürlich wiederholt....)

- Primfaktoren die mehrfach enthalten sind, werden auch mehrfach (untereinander) ausgegeben. Auch nach dem letzten Primfaktor kommt ein Zeilenumbruch.
- Das Programm gibt im Falle einer Primzahl keine Primfaktoren sondern den Text "... ist prim." (mit Zeilenumbruch) aus:

---

```
printf ("%d_ist_prim.\n", ...);
```

---

- Das Programm gibt im Fehlerfall ( $z < 2$ ) den Text "... kann nicht zerlegt werden." (mit Zeilenumbruch) aus:

---

```
printf ("%d_kann_nicht_zerlegt_werden.\n", ...);
```

---

- Weitere Ein- und Ausgaben dürfen nicht erfolgen.
- Das Programm darf keine Felder enthalten, Funktionen sind (zusammen mit erläuterten Prototypen) erlaubt.
- Testen Sie Ihr Programm bitte auch mit großen (Prim-) Zahlen als Eingabe!!!!
- Das Programm ist mit dem Dateinamen <matrikel-nr>-testat-3.c (also bspw. 12345-testat-3.c) bei moodle hochzuspielen, Termin ist 9.11.2019 8:30 Uhr

Für das obige Beispiel 6936 wäre die Ausgabe also

```
2
2
2
3
17
17
```