Aufgabe 11

Erstellen Sie ein Programm, das sich zufällig eine Zahl aus dem Intervall [1,x] denkt. X ist dabei vom Benutzer einzugeben. Danach soll der Benutzer versuchen, die vom Computer gedachte Zahl zu erraten. Für jeden Rateversuch wird dem Benutzer mitgeteilt, ob seine Zahl zu groß oder zu klein ist.

Beispiel für einen möglichen Ablauf:

```
Du musst versuchen eine Zahl zu erraten, wie groß darf die zu
erratende Zahl maximal sein: 40

Ok. Ich habe die Zahl.
   1. Versuch: 20
      ....zu niedrig
   2. Versuch: 30
      ... zu hoch
   3. Versuch: 25
      ... zu hoch
   4. Versuch: 22
      ... richtig
Du hast 4 Versuche zum Erraten der Zahl benötigt
```

Aufgabe 12

Schreiben Sie ein Programm, das die Quersumme einer eingegebenen Zahl berechnet und ausgibt.

Aufgabe 13

Schreiben Sie ein Programm, welches eine eingelesene natürliche Zahl in ihre Primfaktoren zerlegt und diese auf dem Bildschirm anzeigt. Die Primfaktorzerlegung soll beispielsweise für die Zahl 18 die folgende Ausgabe liefern: `18 = 2 * 3 * 3`

Für die Realisierung dürfen Sie ein einfaches Verfahren anwenden. Testen Sie beispielsweise einfach für alle möglichen Teiler von 2 bis zur Zahl selbst, ob ein ganzzahliges Teilen ohne Rest möglich ist. Falls Sie einen Teiler gefunden haben, dann teilen Sie und wiederholen das Verfahren entsprechend.

Aufgabe 14

Lesen Sie solange Strings zeilenweise von der Tastatur ein und geben Sie diese umgekehrt aus. Das Abbruchkriterium ist STRG+D (STRG+Z)

Aufgabe 15

Lesen Sie eine Anzahl von Zahlen ein. Bestimmen Sie die größte und die kleinste Zahl und geben Sie diese aus.
Geben Sie die Zahlen sortiert aus.

Aufgabe 16

Gegeben sei das folgende Array:
@song = qw (They sentenced me to twenty years of boredom);

Schreiben Sie ein Programm das zeilenweise einen oder mehrere Indizes einliest. Die zugehörigen Elemente sollen dann zeilenweise getrennt ausgegeben werden. Fangen Sie ungültige Eingaben ab!

Aufgabe 17

Sortieren Sie dieses Array nach den ASCII Werten:
@song = qw (They sentenced me to twenty years of boredom);

Aufgabe 18

a) Initialisieren Sie ein Array mit 10 Benutzern und geben Sie es in einer Schleife von vorne nach hinten und von hinten nach vorne aus.

b) Legen Sie zusätzlich ein Array an, das für jeden Benutzer ein eigenes Passwort enthält.
Dieses Array sollte zum Array der Benutzernamen parallel sein, d.h. steht der Benutzername an Stelle i im ersten Array, so steht das zugehörige Passwort auch an Stelle i.
Geben Sie die Paare bestehend aus Benutzername und zugehörigem Passwort aus.

c) Lesen Sie einen Benutzernamen ein und suchen Sie den Benutzer im Array. Falls Sie ihn nicht finden, geben Sie eine Fehlermeldung aus.

d) Simulieren Sie das Login-Verfahren mit den beiden Arrays. Fragen Sie nach Login und Passwort und prüfen Sie, ob das Passwort zum Benutzer passt.

e) Ermöglichen Sie durch ein drittes Array eine individuelle Anzahl von Login-Versuchen für jeden Benutzer. Fragen Sie also zuerst nach dem Benutzernamen und dann höchstens so oft nach dem Passwort, wie der Eintrag des dritten Array angibt.