

### Aufgabe 1: (Fasnets-App)

Passend zur Fastnacht (oder Fasnet, Fasching, Karneval...) folgende Aufgabe:

*(Hat ausnahmsweise mit enum und struct nichts zu tun...)*

Der Benutzer wählt, wo er feiern möchte, das Programm antwortet mit dem regional passenden Narrengruß und dem passenden (?) Getränk, etwa so:

#### Aufgabe 1: Fasnetsprogramm

Wo feiern Sie:

- 1: Koeln
- 2: Duesseldorf
- 3: Konstanz
- 4: Egal - es ist Aschermittwoch... 1

Alaaf! Wir trinken ein Koelsch!

Wo feiern Sie:

- 1: Koeln
- 2: Duesseldorf
- 3: Konstanz
- 4: Egal - es ist Aschermittwoch... 2

Helau! Wir trinken ein Alt!

Wo feiern Sie:

- 1: Koeln
- 2: Duesseldorf
- 3: Konstanz
- 4: Egal - es ist Aschermittwoch... 3

Narri Narro! Wir trinken Moscht!

Wo feiern Sie:

- 1: Koeln
- 2: Duesseldorf
- 3: Konstanz
- 4: Egal - es ist Aschermittwoch... 4

Das Programm wird in einer Schleife beliebig oft durchlaufen.

An Aschermittwoch (Benutzer-Eingabe der Zahl 4) wird das Programm jedoch beendet.

Pädagogischer Zusatz-Hinweis:

Es geht natürlich auch ohne Alkohol...

## Aufgabe 2 (Strukturen):

Sie erinnern sich sicher unserer Dönerbude!?!?

Wir haben verschiedene Speisen angeboten, Bestellungen entgegen genommen und den Gesamtpreis ausgerechnet, etwa so:

```
Willkommen in unserer Doenerbude!

Wir koennen Ihnen heute 3 Speisen anbieten:
1. Lahmacun           Preis: 3.75 EUR
2. Doener-Kebap      Preis: 3.45 EUR
3. Pommes-Frites     Preis: 2.50 EUR

Wieviele Lahmacun moechten Sie kaufen?  2
Wieviele Doener-Kebap moechten Sie kaufen? 1
Wieviele Pommes-Frites moechten Sie kaufen? 2

Besten Dank fuer Ihre Bestellung, die wir wie folgt ausfuehren:
2 * Lahmacun,          kostet zusammen 7.50 EUR
1 * Doener-Kebap,     kostet zusammen 3.45 EUR
2 * Pommes-Frites,    kostet zusammen 5.00 EUR

Gesamtwert Ihrer Bestellung: 15.95 EUR. Guten Appetit!

Drücken Sie eine beliebige Taste . . .
```

Lösen Sie die Aufgabe nun unter Verwendung von einer (1) Struktur namens „Angebot“, diese Struktur enthält den Namen der angebotenen Speise und natürlich den Einzelpreis.

## Aufgabe 3 (enum):

Erstellen Sie ein Programm, das eine Ampelschaltung simuliert.

Definieren Sie für die Darstellung der verschiedenen Ampelzustände (rot, rot-gelb, grün, gelb) einen eigenen enum-Typ **Ampel**.

Beginnen Sie bei der Ampelstellung „rot“ und schalten Sie dann die Ampel mit einer Tastenbetätigung immer weiter, programmieren Sie dazu eine „Endlosschleife“. Geben Sie jeden Ampelzustand mit einer Meldung am Bildschirm aus.

Wegen der Endlosschleife können Sie das Programm nur „hart“ beenden, also das Fenster direkt schließen...

```
Meine Ampel ist jetzt rot
Drücken Sie eine beliebige Taste . . .
Meine Ampel ist jetzt rot-gelb
Drücken Sie eine beliebige Taste . . .
Meine Ampel ist jetzt gruen
Drücken Sie eine beliebige Taste . . .
Meine Ampel ist jetzt gelb
Drücken Sie eine beliebige Taste . . .
Meine Ampel ist jetzt rot
Drücken Sie eine beliebige Taste . . .
Meine Ampel ist jetzt rot-gelb
Drücken Sie eine beliebige Taste . . .
Meine Ampel ist jetzt gruen
Drücken Sie eine beliebige Taste . . .
Meine Ampel ist jetzt gelb
Drücken Sie eine beliebige Taste . . .

( und so weiter bis in alle Ewigkeit...)
```