

# PROGC Selbststudium 2 - Kontrollstrukturen

## Inhalt

PROGC Selbststudium 2 - Kontrollstrukturen .....	1
1 Berechnung des Folgetages.....	1
1.1 Aufgabe.....	1
1.2 Hinweise .....	2
2 Korrektheit des eingegebenen Datums.....	2
2.1 Aufgabe.....	2
3 Testen der Aufgaben.....	2

## 1 Berechnung des Folgetages

### 1.1 Aufgabe

Schreiben Sie ein C-Programm, welches von einem eingegebenen gültigen Datum das Datum des folgenden Tages berechnet und ausgibt.

Verwenden Sie Aufzählungstyp (**enum**), um die Monatsnamen *Jan*, *Feb*,..., *Dec* als konstante **int**-Werte *1...12* zu definieren. Arbeiten Sie dann im Programm immer mit den Monatsnamen, wenn Sie die Monatsnummern 1 – 12 verwenden wollen.

Deklariert Sie den Aufzählungstyp am Anfang des Programms, direkt nach den **#include**-Direktiven (damit der Aufzählungstyp im ganzen Programm bzw. in allen Funktionen verwendet werden kann), wie folgt:

```
enum {Jan=1, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec};
```

Ausserdem soll ein Datum bestehend aus Tag, Monat und Jahr als Struktur (**struct**) aufgefasst werden. Deklarieren Sie diese Struktur im Programm gleich nach dem obigen Aufzählungstyp (damit auch die Struktur im ganzen Programm verwendet werden kann) wie folgt:

```
typedef struct {
    int day;
    int month;
    int year;
} Date;
```

Wir haben für die Struktur mit **typedef** bei der Deklaration gleich noch einen Alias dafür definiert. Dadurch können Variablen ganz einfach mit z.B. **Date start\_date;** definiert werden.

Packen Sie die Berechnung des Folgetages in eine eigene Funktion. Verwenden Sie weitere Funktionen, wenn dies für Sie sinnvoll erscheint. Überlegen Sie sich bevor Sie mit dem eigentlichen Programmieren beginnen, wie das Programm in mehrere Funktionen strukturiert werden soll. Eine sinnvolle Struktur erhalten Sie zum Beispiel, indem Sie das Datum in der main-Funktion einlesen und anschliessend einer anderen Funktion übergeben, welche den Folgetag des eingegebenen Datums berechnet.

## 1.2 Hinweise

- Berücksichtigen Sie Schaltjahre des bei uns gültigen Gregorianischen Kalenders. Ein Jahr ist ein Schaltjahr, wenn es durch 4 aber nicht zugleich durch 100 teilbar ist. Ist das Jahr durch 400 teilbar, dann ist es aber wieder ein Schaltjahr.
- Nur Daten ab 1. Januar 1583 sollen auf Ihre Gültigkeit geprüft werden (das ist das Jahr nach der gregorianischen Kalenderreform).
- Verwenden Sie die Funktion `scanf` aus der Standard Library um das Datum von der Tastatur einzulesen. Wenn `day`, `month` und `year` die Variablen vom Typ `int` für das einzulesende Datum sind, wird `scanf` folgendermassen verwendet:

```
scanf("%d%d%d", &day, &month, &year);
```

Die Eingabe auf der Kommandozeile ist dann z.B. mit `24 12 2004`<return> möglich.

## 2 Korrektheit des eingegebenen Datums

### 2.1 Aufgabe

Prüfen Sie bei der Eingabe, ob das eingelesene Datum gültig ist.

## 3 Testen der Aufgaben

Verwenden Sie wie in den Praktika das `CUnit` Test Framework. Lesen Sie nochmals die Anweisungen im Dokument `SelbstStudium01-Kontrollstrukturen.pdf`.

Definieren Sie Unit Tests für die einzelnen Funktionen sowie Tests welche von der Kommandozeile Daten einliest und den Output generiert.