

Übungsblatt 4

Aufgabe 1: Bedingte Ausführung

Ein Händler gibt auf den Verkauf von DVD-Rohlingen bei Abnahme höherer Stückzahlen Rabatt. Ein Rohling kostet 0,80 Euro. Bei Verkauf ab 10 Stück gibt er 3% Rabatt, ab 50 Stück 5% und ab 100 Stück 8%.

Schreiben Sie ein Programm, das die gewünschte Stückzahl einliest und anschließend den Verkaufspreis ausgibt. Runden Sie bei der Ausgabe des Preises auf ganze Cent-Beträge (kaufmännisches Runden).

Aufgabe 2: Bedingte Ausführung

Im Jahre 1582 wurde der gregorianische Kalender eingeführt. Mit Hilfe der folgenden Formel lässt sich zu jedem Datum ab dem Jahr 1582 der zugehörige Wochentag w ermitteln, wobei dem Sonntag die 0, dem Montag die 1, dem Dienstag die 2, usw. entspricht:

$$w = (tag + 2 * m + (3 * m + 3) / 5 + j + j / 4 - j / 100 + j / 400 + 1) \% 7, \text{ wobei}$$

$$m = \begin{cases} monat + 12, & \text{falls } monat \leq 2 \\ monat, & \text{sonst} \end{cases}$$

$$j = \begin{cases} jahr - 1, & \text{falls } monat \leq 2 \\ jahr, & \text{sonst} \end{cases}$$

Schreiben Sie ein C-Programm, das einen Tag, ein Monat und ein Jahr einliest und dann die Berechnung des Wochentags nach obiger Formel durchführt. Anschließend soll eine Ausgabe der folgenden Form erzeugt werden:

Der 1. 11. 2010 ist ein Montag

Aufgabe 3: Bitoperatoren und bedingte Ausführung

In dieser Aufgabe sollen Sie einen Algorithmus erstellen, der die **schriftliche Addition zweier binärer 2-Bit Zahlen** durchführt. Als Eingabe bekommen Sie zwei 8-Bit Zahlen (gegeben in zwei Variablen vom Typ **unsigned char**) und zur Ausgabe speichern Sie das Ergebnis in einer Variable vom Typ **unsigned short int**.

Gehen Sie hierbei Bit-weise vor! Das heißt: extrahieren Sie jeweils die Bits (Operanden) aus den Eingabe-Variablen und speichern Sie das Ergebnis der Bit-Addition und den Übertrag in Zwischenvariablen bevor Sie das Ergebnis-Bit an die richtige Stelle in der Ausgabe-Variable schreiben.

Wichtig: Sie benötigen für die Umsetzung der Addition **keine arithmetische Operation** (z.B. +). Die Addition zweier Bit soll über Bedingte Ausführung umgesetzt werden.

Eingabe über cin:

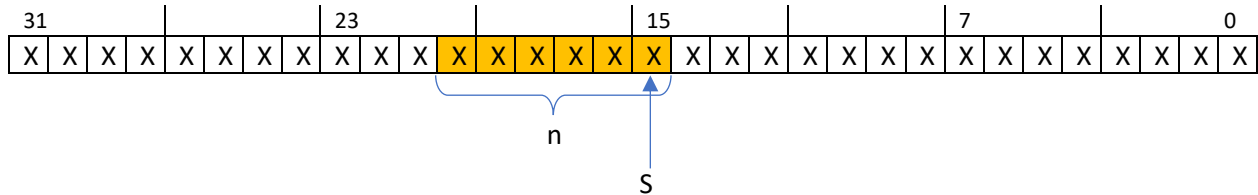
Beachten Sie bei einer Eingabe, dass Werte eines Datentypen **unsigned char** immer als Zeichen erwartet werden. Wie können Sie es erreichen, dass Sie dem Programm Ihre gewünschten Zahlwerte mittels eines **cin**-Befehls übergeben können?

Eingabe über scanf:

Beachten Sie bei einer Eingabe, dass Werte eines Datentypen **unsigned char** über format specifier **%c** eingelesen werden können, diese aber als Zeichen erwartet werden. Wie können Sie es erreichen, dass Sie dem Programm Ihre gewünschten Zahlwerte mittels eines **scanf**-Befehls übergeben können?

Aufgabe 4: Bitoperatoren ohne bedingte Ausführung

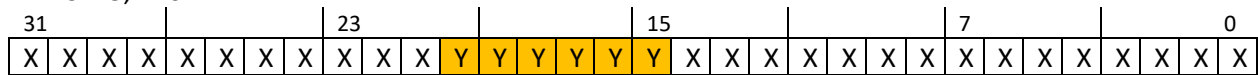
In dieser Aufgabe sollen Sie 4 Programme entwickeln die zur Bit-Manipulation eines Konfigurations-Registers benutzt werden können. Als Eingabe des Programms sollen die Werte **S** und **n** eingelesen werden. Das Programm soll auf einem 32 Bit Integer-Wert arbeiten.



Bedingte Ausführung mittels **if** oder Schleifen u.ä. sind **nicht** zu verwenden.

1. Erstellen Sie ein Programm, dass n Bits startend ab Bit-Position S extrahiert

z.B.: S=15, n=6



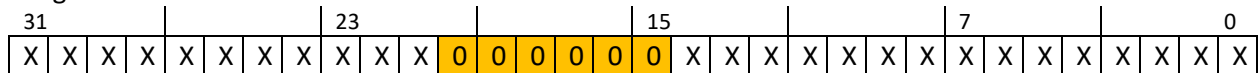
Ausgabe:



2. Erstellen Sie ein Programm, dass n Bits startend ab Bit-Position S löscht (Wert == 0)

z.B.: S=15, n=6

Ausgabe:



3. Erstellen Sie ein Programm, dass alle bitts außer n Bits startend ab Bit-Position S löscht (Wert == 0)

z.B.: S=15, n=6

Ausgabe:



4. Erstellen Sie ein Programm das eine Eingabe der Größe n an die Bit Position S schreibt

z.B.: S=15, n=6, Value = Z



Ausgabe:

