

# Klausur Programmieren 1

HAW-Hamburg, Fakultät Technik und Informatik, Department Informations- und Elektrotechnik

Prof. Dr. Robert Heß, 21.1.2019, Bearbeitungsdauer: 180 Min.

Erlaubte Hilfsmittel: Vorlesungsunterlagen, Lösungen aus dem Praktikum und C/C++ Einführungsbücher.

Ergebnis: ..... von 100 Punkten

Note: ..... Punkte.

## 1 Einleitung

Neben den festen Feiertagen wie *Tag der deutschen Einheit* oder *1. Weihnachtstag* gibt es hierzulande einige Feiertage, die jedes Jahr auf unterschiedliche Daten fallen. Der Abstand zwischen diesen Tagen ist dabei fix. Die folgende Tabelle zeigt die beweglichen Feiertag mit den Abständen zu Ostersonntag und beispielhaft die Daten für die Jahre 2018, 2019 und 2020:

Feiertag	rel. zu Ostersonntag	2018	2019	2020
Karfreitag	2 Tage davor	30.3.	19.4.	10.4.
Ostersonntag	—	1.4.	21.4.	12.4.
Ostermontag	1 Tag danach	2.4.	22.4.	13.4.
Himmelfahrt	39 Tage danach	10.5.	30.5.	21.5.
Pfingstsonntag	49 Tage danach	20.5.	9.6.	31.5.
Pfingstmontag	50 Tage danach	21.5.	10.6.	1.6.

Zur Datierung der Feiertage muss von daher nur das Datum des Ostersonntags bestimmt werden. Für den im Westen üblichen gregorianischen Kalender wurde dafür von Carl Friedrich Gauß die sog. *Gaußsche Osterformel* entwickelt, die später noch leichte Korrekturen erfuhr. Mit dieser wird der Tag *OS* im Monat März bestimmt, auf den der Ostersonntag fällt. Werte über 31 bedeuten, das Ostern in den April fällt (z.B. entspricht *OS* = 32 dem 1.4.). Die Formel lautet:

$K := \text{Jahr} \text{ div } 100$	Säkularzahl
$M := 15 + (3 \cdot K + 3) \text{ div } 4 - (8 \cdot K + 13) \text{ div } 25$	säkulare Mondschaltung
$S := 2 - (3 \cdot K + 3) \text{ div } 4$	säkulare Sonnenschaltung
$A := \text{Jahr} \text{ mod } 19$	Mondparameter
$D := (19 \cdot A + M) \text{ mod } 30$	Keim für ersten Vollmond im Frühling
$R := (D + A \text{ div } 11) \text{ div } 29$	kalendarische Korrekturgröße
$OG := 21 + D - R$	Ostergrenze
$SZ := 7 - (\text{Jahr} + \text{Jahr} \text{ div } 4 + S) \text{ mod } 7$	erster Sonntag im März
$OE := 7 - (OG - SZ) \text{ mod } 7$	Entfernung Ostersonntag zu Ostergrenze
$OS := OG + OE$	Tag des Ostersonntags im März

Dabei bedeutet *div* die ganzzahlige Division unter Vernachlässigung der Nachkommastellen, also in C die Division von ganzen Zahlen. Der Operator *mod* liefert den Rest einer Division, also in C der Operator *%*.

## 2 Programmieraufgaben

### Aufgabe 1 (5 Punkte)

Erstellen Sie eine Funktion, die eine ganze Zahl mit Unter- und Obergrenze fehlerfrei vom Benutzer abfragt.

### Aufgabe 2 (20 Punkte)

Erstellen Sie eine Funktion mit Namen *Osterformel*, die als einzigen Parameter eine Jahreszahl erwartet und über *return* den Tag des Ostersonntags im März gemäß Osterformel zurück gibt. Hier ein paar Ergebnisse zum Testen:

Jahr:	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Osterformel:	35	55	39	31	51	36	27	47	32	52	43

### Aufgabe 3 (15 Punkte)

Erstellen Sie eine Funktion mit Namen *Feiertage* mit Rückgabetyp *void*, die als einzigen Parameter eine Jahreszahl erwartet. Die Funktion ermittelt mittels der Funktion aus Aufgabe 2 die Daten aller Feiertage aus der Tabelle in der Einleitung mit Tag und Monat, und gibt diese ordentlich auf dem Bildschirm aus. Beachten Sie, dass Himmelfahrt und Pfingsten auch im Mai oder Juni liegen können.

### Aufgabe 4 (12 Punkte)

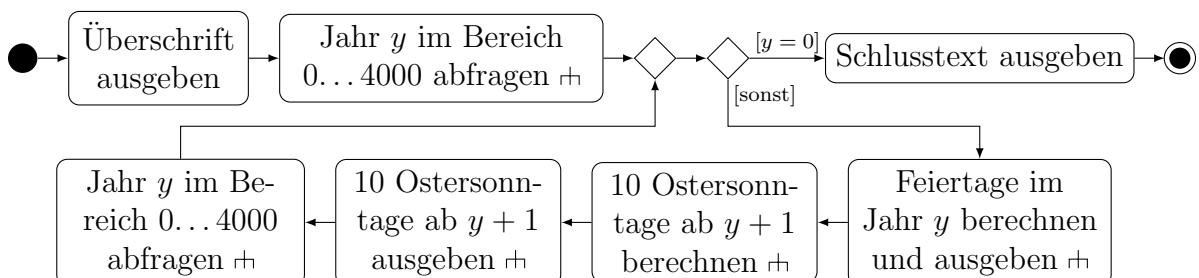
Erstellen Sie eine Funktion mit Namen *OsternBerechnen*, welche die Daten der Ostersonntage für zehn aufeinander folgende Jahre mit Tag und Monat bestimmt und in Vektoren speichert. Die Funktion erwartet als Parameter eine ganze Zahl für das erste der zehn Jahre, einen Vektor mit zehn ganzen Zahlen für die Tage und einen weiteren Vektor mit zehn ganzen Zahlen für die Monate. Die Funktion führt nur die Berechnung durch und gibt nichts auf dem Bildschirm aus.

### Aufgabe 5 (8 Punkte)

Erstellen Sie eine Funktion mit Namen *OsternAusgeben*, welche die Daten der Ostersonntage für zehn aufeinander folgende Jahre auf dem Bildschirm ausgibt. Die Funktion erwartet als Parameter eine ganze Zahl für das erste der zehn Jahre, einen Vektor mit zehn ganzen Zahlen für die Tage und einen weiteren Vektor mit zehn ganzen Zahlen für die Monate. Die Funktion führt nicht die Berechnung aus, sondern gibt nur die übergebenen Daten auf dem Bildschirm aus.

### Aufgabe 6 (20 Punkte)

Fügen Sie die Programmstücke zu einem lauffähigen Programm zusammen:



Laden Sie am Ende nur die Quellcodes (mit Endungen .c und .h) in das für Sie vorbereitete Verzeichnis unter: L:\Prof\abf081\Stud\...

### 3 Verständnisfragen

#### Aufgabe 7 (4 Punkte)

Nennen Sie jeweils einen Vorteil von Programmiersprachen die auf a) Übersetzern und b) Interpretierern basieren.

#### Aufgabe 8 (3 Punkte)

Bestimmen Sie das Bitmuster der Zahl  $-6$  für ein Byte.

#### Aufgabe 9 (2 Punkte)

Definieren Sie einen Vektor mit 11 Variablen vom Typ *short* mit Namen *Zahl*.

#### Aufgabe 10 (6 Punkte)

Welchen Datentyp haben folgende Ausdrücke?

1./2	
exp(0)	
1/2*2	

01234	
1	
”1”	

#### Aufgabe 11 (2 Punkte)

Stellen Sie von der Variablen **unsigned Zahl** die niederwertigsten sechs Bits dezimal dar.

printf(”%u”, );

#### Aufgabe 12 (3 Punkte)

Die unten gezeigte Programmzeile soll nur dann in das Programm eingebunden werden, wenn das Makro **\_RELEASE** nicht definiert wurde. Ergänzen Sie den Code.

printf(”Wert\_der\_Variable\_x:\_%d\n”, x);