

Aufgabe 1

- a) Kreuzen Sie die Definition(en) an, bei denen es sich um ein Makro mit einem Parameter handelt:

- ☐ `#define DRUCK (z) printf("%i", z);`
☐ `#define DRUCK(z) printf("%i", z);`
☐ `#define DRUCK z printf("%i", z);`

- b) Was wird von dem folgenden Programm ausgegeben (Hinweis: Verwenden Sie die ASCII-Tabelle und achten Sie auf die unterschiedlichen Formatangaben):

```
#define a '<'  
#define b 'a'  
#define c 0x0D  
#define d(x) (b+x)  
  
void main()  
{  
    printf("%c%c%x%c%i", a+c, d(c), 15, b+14, a/22);  
}
```

Ausgabe:

- c) Welche Werte haben die Variablen a und b **nach** Ausführung des folgenden Code-Fragments. Kreuzen Sie die richtige Antwort an:

```
int a = 1, b = 1;  
b = a++ + ++b;
```

- ☐ a == 1 und b == 2
☐ a == 2 und b == 3
☐ a == 2 und b == 4

- d) Welchen Wert hat m, wenn x = 3 und y = 3 ist, nach den Anweisungen

```
if (x != y) m = 2*x;  
else      m = 3*y;
```

Aufgabe 2

a) Für den Aufruf einer Funktion benötigt der Compiler (kreuzen sie die richtige Antwort an):

- ☐ keine weiteren Informationen
- ☐ die **Deklaration** der Funktion durch einen Funktionsprototypen
- ☐ die **Definition** der Funktion durch Implementierung der Anweisungen

b) Nennen Sie die Funktion, die nach Ausführung eines C-Programms immer zuerst ausgeführt wird?

c) Die Elemente eines Vektors belegen immer einen zusammenhängenden Speicherbereich?

- ☐ Richtig
- ☐ Falsch

d) Für einen beliebigen Vektor v und einen Index i sind die Ausdrücke $\&v[i]$ und $v+i$ äquivalent?

- ☐ Richtig
- ☐ Falsch

e) Die Zeichenkette "Informatik2" belegt _____ Bytes.

f) Nach Ausführung des Code-Fragments:

```
int i = 0, var = 100;  
if (i != 0) var *= 2;  
else      var /= 2;
```

hat die Variable var den Wert: _____

g) Nach Ausführung der Schleife:

```
int i, sum;  
for (sum = 0, i = 1; i < 5; ++i) sum += i;
```

hat die Variable sum den Wert: _____

Aufgabe 3

- a) Definieren Sie ein Array für 8 ganzzahlige Werte und befüllen Sie es mit Hilfe einer Schleife mit den Quadratzahlen von 1 - 8, beginnend mit 1^2 , 2^2 , usw. (also 1, 4, 9, ...).

- b) Was wird durch die folgende for()-Schleife ausgegeben?

```
int i, *iptr = &i;

for (i=1; i<30; i++) {
    printf("%i, ", *iptr);
    i += i;
}
```

Ausgabe:

Aufgabe 4

- a) Erstellen Sie eine Enumeration namens **Position** mit den folgenden Ausprägungen: **Torwart**, **Verteidigung**, **Mittelfeld**, **Angriff**.
- b) Erstellen Sie eine C-Struktur namens **Spieler**, die sinnvolle Datentypen (inklusive eventueller Bitfelder) für die Unterelemente **name**, **verein** (jeweils 30 Zeichen), **nummer** (Werte von 1 bis 31), **position** (Werte von 0..3, siehe Aufgabe a)) und **gel_km** (gelaufene Kilometer als Fließpunktzahl) enthält!
- c) Erzeugen Sie damit ein Variablenfeld namens **team** für 11 Einträge und weisen Sie dem 11. Eintrag den folgenden Datensatz zu:

name:	"Thomas Müller"
verein:	"Bayern München"
nummer:	13
position:	Angriff
gel_km:	10.7

Aufgabe 4 (Fortsetzung)

- d) Vervollständigen Sie die unten angegebene Funktion so, dass diese die Ausgabe eines Spieler-Datensatzes auf dem Bildschirm wie in der Abbildung erzeugt:

```
Name:      Thomas Müller
Verein:    Bayern München
Nummer:    13
Position:  Angriff
Gel. km:   10.70
```

Hinweis: Unterscheiden Sie beim Unterelement **position** zwischen dem definierten Datentyp und der gewünschte Ausgabe und nutzen Sie dafür zum Beispiel eine **switch-case** Anweisung.

```
void printSpieler(struct spieler *sp)
{

}

}
```

Angaben und Informationen zu den Aufgaben 5-8

Ein Vitalparameter-Messgerät sei an einen Rechner angeschlossen. Dieses liefert fortlaufend Werte für Körpertemperatur, Puls und Blutdruck (systolisch und diastolisch) eines Patienten und speichert diese zu jeder Minute in einer Textdatei ab.

Bei den folgenden Aufgaben 5-7 sind Funktionen zu schreiben, die dann von einer in Aufgabe 8 zu erstellenden `main()`-Funktion aufgerufen werden.

Die Funktionen dienen zur Umrechnung von Temperatureinheiten, zum Einlesen der Messwerte sowie der Mittelwertbildung der Messwerte. Ziel soll es am Ende sein, von jeweils 10 Messwerten aus der Datei Mittelwerte zu bilden und diese auf dem Bildschirm auszugeben.

Zu jeder Funktion sind die dazu notwendigen Header-Dateien (Bibliotheken) sowie der **Funktionsprototyp** anzugeben. Geben Sie nur die **notwendigen Header-Dateien** an. Überzählige bzw. nicht notwendige Header-Dateien führen ggf. zu Punktabzug.

Der Einfachheit halber wird mit **typedef** ein neuer Struktur-Datentyp definiert, der für **eine** Messung alle relevanten Werte zusammenfasst. Dieser soll in den folgenden Aufgaben 6-8 benutzt werden:

```
typedef struct _messung
{
    float temp;           // Temperatur
    float puls;           // Puls
    float sys;            // systolischer Blutdruck
    float dia;            // diastolischer Blutdruck
} Messung;
```

Aufgabe 5

Die Temperaturangaben des Vitalparameter-Messgerätes sind leider in Grad Fahrenheit. Schreiben Sie eine C-Funktion zur Konvertierung einer Temperatur von Grad Fahrenheit in Grad Celsius. Der Fahrenheitwert wird als Fließpunktzahl mit einfacher Genauigkeit als einziger Parameter dieser Funktion übergeben. Der in Grad Celsius umgewandelte Wert wird von der Funktion zurückgeliefert.

Für die Umrechnung gilt: `celsius = 5/9*(fahrenheit - 32)`.

Notwendige Header-Dateien:

Funktionsprototyp:

Funktionsdefinition:

Aufgabe 6

Schreiben Sie eine C-Funktion zum Einlesen mehrerer Messwerte von einer Datei. Dazu werden dieser Funktion drei Parameter übergeben: der Dateizeiger vom Typ **FILE***, ein Array vom Typ **Messung** (siehe Seite 7) sowie die Anzahl der Datensätze (Zeilen) die einzulesen sind.

Nehmen Sie dabei an, dass die Werte für Temperatur, Pulse und Blutdruck (systolisch und diastolisch) pro Datensatz zeilenweise wie folgt in der Textdatei stehen (siehe Beispiel-Abbildung rechts):

99.2	68	120	80
99.4	69	121	79
99.5	71	123	81
99.6	69	119	79
99.4	68	117	79
99.3	67	118	85
99.2	68	120	80

<Temperatur> <Puls> <systolisch> <diastolisch>

Die Temperaturwerte sind dort als Fließpunktzahlen, die anderen Werte als Ganzzahlen abgespeichert. Hinweis: Achten Sie auf die Datentypen der Elemente der Struktur **Messung** (siehe Seite 7).

Notwendige Header-Dateien:

Funktionsprototyp:

Funktionsdefinition:

Aufgabe 7

Schreiben Sie eine C-Funktion zur Berechnung des arithmetischen Mittels der einzelnen Werte aus einer beliebigen Anzahl von Messwert-Datensätzen. Jeder Messwert-Datensatz besteht dabei aus einem Eintrag vom Typ **Messung** (siehe Seite 7), d.h. von den Messungen sind zu bestimmen der Mittelwert für Temperatur, Pulse und Blutdruck (systolisch und diastolisch). Der Funktion werden die Messungen als Array vom Typ **Messung** und die Anzahl der Messungen als Ganzzahl übergeben. Die berechneten Mittelwerte werden über eine Struktur vom Typ **Messung** zurückgegeben.

Das arithmetische Mittel $\langle x \rangle$ einer Anzahl von N Werten ist: $\langle x \rangle = \frac{1}{N} \sum_{i=1}^n x_i$

Notwendige Header-Dateien:

Funktionsprototyp:

Funktionsdefinition:

Aufgabe 8

Schreiben Sie eine Hauptfunktion, die eine Text-Datei mit Namen "**vitals.txt**" zum **Lesen** öffnet. Achten Sie dabei auf korrekte Dateibehandlung, falls diese nicht existiert oder nicht geöffnet werden kann. Lesen Sie dann mit Hilfe der in Aufgabe 6 geschriebenen Funktion **solange** jeweils **10 Messwert-Datensätze** ein, bis das Dateieinde erreicht ist. Bestimmen Sie dann immer von diesen 10 Messwerten mit Hilfe der in Aufgabe 7 geschriebenen Funktion die Mittelwerte für Temperatur, Puls und Blutdruck (systolisch und diastolisch) und geben diese wie untenstehend auf dem Bildschirm aus.

Beispielausgabe:

```
<T> = 37.4 C    <Puls> = 69.3 1/min    <Blutdruck> = 121.3/80.0 mmHg  
<T> = 37.5 C    <Puls> = 68.0 1/min    <Blutdruck> = 118.0/81.0 mmHg
```

Notwendige Header-Dateien:

Funktionsdefinition: