Aufgabe 1: (8 Punkte)

Bei den Multiple-Choice-Fragen ist jeweils nur **eine** richtige Antwort eindeutig anzukreuzen. Auf die richtige Antwort gibt es die angegebene Punktzahl.

Wollen Sie eine Multiple-Choice-Antwort korrigieren, kreisen Sie bitte die falsche Antwort ein und kreuzen die richtige an.

Lesen Sie die Frage genau, bevor Sie antworten.

a) Welche der folgenden Aussagen bzgl. der Interruptsteuerung ist richtig? 2 Punkte

- ❏ Pegel-gesteuerte Interrupts werden beim Wechsel des Pegels ausgelöst, daher der Name.
- ❏ Pegel-gesteuerte Interrupts werden immer wieder ausgelöst, solange ein bestimmter Pegel anliegt.
- ❏ Während flankengesteuerte Interrupts blockiert sind, können trotzdem pegelgesteuerte Interrupts der gleichen Interrupt-Quelle noch eintreffen und bearbeitet werden.
- ❏ Wurde gerade ein pegelgesteuerter Interrupt ausgelöst, so muss erst ein Pegelwechsel der Interruptleitung stattfinden, bevor erneut ein Interrupt ausgelöst wird.

b) Was versteht man unter Polling? 2 Punkte

- ❏ Wenn ein Gerät durch Auslösen eines Interrupts Daten von einem Mikrocontroller anfordert.
- ❏ Wenn ein Programm regelmäßig eine Peripherie-Schnittstelle abfragt, ob Daten oder Zustandsänderungen vorliegen.
- ❏ Wenn ein Programm zum Zugriff auf kritische Daten Interrupts sperrt.
- ❏ Wenn ein Gerät so lange Interrupts auslöst, bis die Daten durch den Mikrocontroller abgeholt wurden.

c) Welcher der folgenden Mechanismen gehört zu den Funktionen eines Betriebssystemkerns wie z. B. UNIX oder Windows? 2 Punkte

- ❏ Entgegennehmen von Kommandos (Shell / Eingabeaufforderung)
- ❏ Verwaltung von Druckaufträgen
- ❏ Interrupt-Behandlung
- ❏ Compiler zum Übersetzen von Programmen

d) Welcher der folgenden Punkte ist kein Attribut einer Datei? 2 Punkte

- ❏ Dateigröße
- ❏ Dateiinhalt
- ❏ Eigentümer-Kennung
- ❏ Zugriffsrechte

Aufgabe 2: (25 Punkte)

a) An einem Port eines Mikrocontrollers sind 8 LEDs angeschlossen. Die einzelnen LEDs können über die Bits eines Registers angesprochen werden. Das Register ist in den Speicher des Microcontrollers auf Adresse 0x3B abgebildet.

Schreiben Sie eine Funktion
`void set_led(int led, int on_off)`
die Folgendes leistet:

Die LED Nr. `led` (Wert von 0 bis 7) wird eingeschaltet (falls `on_off` = 1) bzw. ausgeschaltet (`on_off` = 0).

Sie können davon ausgehen, dass die Funktion nicht mit falschen Parametern aufgerufen wird.

```
/* Funktion set_led */
```

G:

b) Schreiben Sie ein Programm **analyse**, das Messwerte (ganze Zahlen) von der Standardeingabe einliest und jeden Wert zusammen mit der aktuellen Zeit (Sekunden seit Start der Messung) in einem Feld abspeichert. Anschliessend sollen alle Messwerte, die einen bestimmten Grenzwert übersteigen (wird als Argument beim Aufruf des `analyse`-programms übergeben) zusammen mit der zugehörigen Zeit auf der Standardausgabe ausgegeben werden.

Beispiel:

```
Kommando:  analyse 17
Eingabe:   2
           26
           6
           18
           ^D
Ausgabe:   26 (5 sek.)
           18 (12 sek.)
```

Beachten sie dabei Folgendes:

- Die Messwerte sollen zusammen mit der Zeit in einer Struktur **messwert** gespeichert werden.
- Das Feld soll bis zu 1000 Messwerte aufnehmen können. Definieren Sie hierfür ein Makro namens **SIZE**.
- Falls der Benutzer keinen Grenzwert beim Aufruf angibt, soll eine Fehlermeldung ausgegeben werden und das Programm abbrechen.
- Das Einlesen der Werte soll in einer eigenen Funktion
  int read_input(struct messwert feld[], int feldlen);
  erfolgen.
  Die Funktion erhält das Feld und dessen Länge als Parameter, liest Werte von der Standardeingabe, speichert sie zusammen mit der aktuellen Zeit in dem Feld und liefert die Zahl der gelesenen Werte als Ergebnis zurück.
- Zum Abfragen der aktuellen Zeit steht Ihnen eine Funktion
  int get_time();
  zur Verfügung, die die Sekunden seit Programmstart zurückliefert.
- Zum Einlesen der Messwerte benutzten Sie am besten die Funktion scanf.
  Schnittstelle: **int scanf(char *format, arg);**
  Das Format-Kürzel **"%d"** liest ganze Zahlen ein und erwartet dazu einen `arg`-Parameter vom Typ **(int *)**.
  Ergebnis ist normalerweise 1, am Ende der Eingabe **EOF**.
- Mit Hilfe der Funktion
  int atoi(char *string);
  können Sie eine als String vorliegende Zahl in einen int-Wert umwandeln.

Sie dürfen dieses Blatt zur besseren Übersicht bei der Programmierung heraustrennen!

Ergänzen Sie das folgende Codegerüst so, dass ein vollständig übersetzbares Programm entsteht.

```
#include <stdio.h>
#include <stdlib.h>
/* Makros, Strukturdeklarationen, Funktiondeklarationen */

/* Funktion main */

   /* Kommandozeilen-Parameter ueberpruefen und auswerten */

   /* Werte einlesen */
```

S:

```
   /* Werte, die Grenzwert übersteigen ausgeben */

   /* Ende der main-Funktion */

/* Funktion read_input
   Werte einlesen und mit Zeitstempel versehen */
```

S:
R:

Aufgabe 3: (12 Punkte)

Die folgenden Beschreibungen sollen kurz und prägnant erfolgen (Stichworte, kurze Sätze)

a) Erläutern Sie die Unterschiede bei der Ausführung eines Programms auf einem Betriebssystem und auf einem "nackten" Mikrocontroller.

b) Was versteht man unter einem Interrupt?

c) Wie wird ein Interrupt typischerweise bearbeitet?