Hochschule
München
University of
Applied Sciences

Wintersemester 2020/21

Ingenieurinformatik, Teil 1: C-Programmierung

Schriftliche Fernprüfung mit Videoaufsicht

Prüfer: Küpper, Reichl und KollegInnen

Bearbeitungszeit: 60 Minuten

Hilfsmittel:

- **Taschenrechner und elektronische Hilfsmittel sind nicht zugelassen.**
- **Alle schriftlichen Unterlagen sind erlaubt.**
- **Der PC darf während der Prüfung nur zur Anzeige des Aufgabenblatts genutzt werden.**

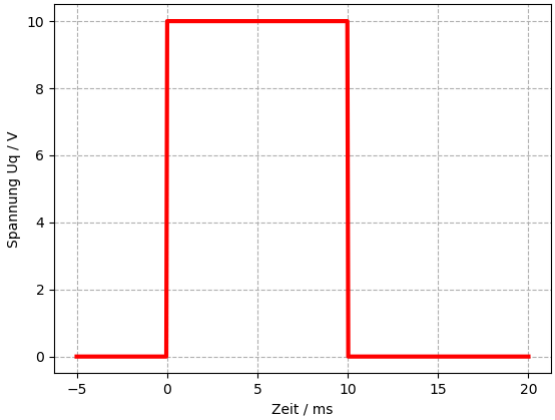
Schreiben Sie Ihren Namen, Vornamen und auch die Studiengruppe auf alle Lösungsblätter. Es werden nur handschriftliche Lösungen auf leeren, weißen DIN-A4-Blättern akzeptiert.

Wenn Sie zur Kombiprüfung „Ingenieurinformatik“ angemeldet sind, dann beachten Sie bitte, dass Sie beide Teile (C-Programmierung und Numerik/ Matlab) im selben Semester schreiben müssen.

****** Viel Erfolg! ******

Aufgabe 1 (allg. Programmierung, ca. 24 Punkte)

Eine Reihenschaltung aus einem Widerstand der Größe $R = 10\Omega$ und einer Spule $L = 0{,}1\mathrm{H}$ ist an eine Spannungsquelle $u_q(t)$ angeschlossen. Die Spannungsquelle $u_q(t)$ wird nur zu Beginn kurz eingeschaltet (siehe Abbildung rechts).



Die Stromstärke $i(t)$ kann folgendermaßen berechnet werden:

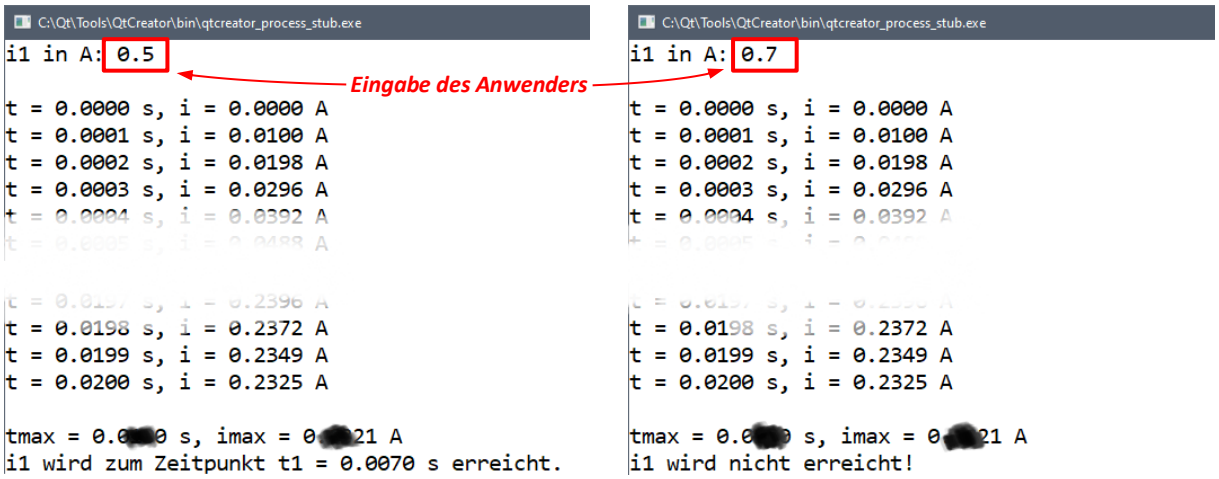
$$i(t) = 1\,\mathrm{A} \cdot \left(1 - \mathrm{e}^{-\frac{t}{0{,}01\,\mathrm{s}}}\right) \quad \text{für} \quad t = 0 \ldots 0{,}01\,\mathrm{s}$$

$$i(t) = 1\,\mathrm{A} \cdot \left(1 - \frac{1}{\mathrm{e}}\right) \cdot e^{-\frac{t - 0{,}01\,\mathrm{s}}{0{,}01\,\mathrm{s}}} \quad \text{für} \quad t > 0{,}01\,\mathrm{s}$$

Schreiben Sie ein C-Programm, welches die folgenden Aufgaben löst:

- Beim Start wird der Anwender nach einem Schwellenwert i_1 für die Stromstärke gefragt. Die Eingabe soll in Ampere (A) erfolgen.
- Die Stromstärke $i(t)$ wird für das Zeitintervall $t = 0 \ldots 0{,}02\,\mathrm{s}$ in Schritten von $\Delta t = 0{,}0001\,\mathrm{s}$ berechnet. Die Zeit in s und die Stromstärke in A sollen jeweils mit vier Nachkommastellen tabellarisch ausgegeben werden.
- Kurz vor dem Ende des Programms, nachdem (!) die Tabelle vollständig auf dem Bildschirm ausgegeben wurde, werden die folgenden Informationen ausgegeben:
  1. Die maximale Stromstärke $i_{\max}$, die im Zeitintervall $t = 0 \ldots 0{,}02\,\mathrm{s}$ geflossen ist und auch der Zeitpunkt $t_{\max}$, wann diese maximale Stromstärke erreicht wurde (jeweils vier Nachkommastellen).
  2. Der Zeitpunkt t_1, wann die Stromstärke zum ersten Mal den Schwellenwert i_1 erreicht hat (vier Nachkommastellen, Abbildung unten links). Es kann auch sein, dass der Schwellenwert gar nicht erreicht wird; in diesem Fall soll eine entsprechende Meldung ausgegeben werden (Abbildung unten rechts).

Die Bildschirmausgabe Ihres Programms soll den folgenden Bildschirmfotos entsprechen:



Aufgabe 2 (Vektoren, Matrizen, Kontrollstrukturen, ca. 16 Punkte)

2.1. In einem C-Programm ist die folgende Matrix als globale Variable definiert:

```
double m[10][10];
```

Es soll eine Funktion mit dem Namen „belegen“ programmiert werden. Diese Funktion schreibt die folgenden Werte in die Matrix: Alle Elemente „auf dem Rand“ der Matrix (also in der ersten und letzten Zeile bzw. in der ersten und letzten Spalte) sollen auf 10 gesetzt werden. Die übrigen Elemente „im Inneren“ der Matrix sollen auf -1 gesetzt werden.

```
10   10   10   10   10   10   10   10   10   10
10   -1   -1   -1   -1   -1   -1   -1   -1   10
10   -1   -1   -1   -1   -1   -1   -1   -1   10
10   -1   -1   -1   -1   -1   -1   -1   -1   10
10   -1   -1   -1   -1   -1   -1   -1   -1   10
10   -1   -1   -1   -1   -1   -1   -1   -1   10
10   -1   -1   -1   -1   -1   -1   -1   -1   10
10   -1   -1   -1   -1   -1   -1   -1   -1   10
10   -1   -1   -1   -1   -1   -1   -1   -1   10
10   10   10   10   10   10   10   10   10   10
```

Zeichnen Sie ein Struktogramm der Funktion „belegen“. Alle verwendeten Kontrollstrukturen müssen klar erkennbar und korrekt gezeichnet sein.

Hinweis: In dieser Teilaufgabe 2.1 soll kein C-Quelltext geschrieben werden!

2.2. In einem C-Programm ist der folgende Vektor als globale Variable definiert:

```
double v[45];
```

Es soll eine Funktion „fibo“ programmiert werden, welche alle (!) Elemente des Vektors der Reihe nach mit den folgenden Werten belegt:

```
1 1 2 3 5 8 13 21 34 55 89 …
```

Es handelt sich dabei um die sog. Fibonacci-Zahlenfolge. Dabei ergibt immer die Summe zweier aufeinanderfolgender Zahlen die nachfolgende Zahl.

Schreiben Sie den C-Quelltext der Funktion „fibo“. Diese Funktion soll die Fibonacci-Zahlen in einer Schleife berechnen und in den Vektor schreiben. (Rekursive Lösungen sind nicht zulässig!)

Hinweis: In dieser Teilaufgabe 2.2 soll kein Struktogramm gezeichnet werden!

Aufgabe 3 (Fehlersuche, Zeiger, ca. 13 Punkte)

Nach dem Programmstart gibt der Anwender zunächst alle 10 Elemente eines double-Vektors ein. Anschließend wird die Funktion „summ_prod“ aufgerufen. Diese Funktion berechnet sowohl die Summe als auch das Produkt von allen Elementen des Vektors.

```
1   #include <stdio.h>
2   #include <math.h>
3
4   | Die Deklaration der Funktion „summ_prod“ fehlt noch … |
5
6
7   int main(void)
8   {
9       double vek[10], summ, prod;
10
11      int anz = 0;
12      while(anz < 10);
13      {
14          printf("Element %d: ", anz + 1);
15          scanf("%f", &vek[anz]);
16
17      }
18
19      summ_prod(vek, anz, &summ, &prod);
20      printf("Summe:   %.1f\n", summ);
21      printf("Produkt: %.1f\n", prod);
22      return 0;
23  }
24
25  | Die erste Zeile der Funktions-Definition fehlt noch … |
26
27  {
28      int n;
29      *summ = 0;
30      *prod = 0;
31
32      for(n = 0; n < anz; n = n++)
33      {
34          *summ += v[n];
35          *prod *= v[n];
36      }
37  }
```

3.1. Die Deklaration und die erste Zeile der Funktions-Definition von „summ_prod“ fehlen noch.

- Wie lautet die Deklaration der Funktion „summ_prod“?
- Wie lautet die erste Zeile der Funktions-Definition von „summ_prod“?

3.2. Im Quelltext befinden sich fünf Fehler.

- In welchen Zeilen befinden sich die Fehler?
- Geben Sie jeweils die korrekte Version der kompletten (!) Zeile an.

Aufgabe 4 (Bits und Bytes, Zeichenketten, ca. 14 Punkte)

4.1. Sie arbeiten mit einem aktuellen C-Compiler auf einem Windows- oder macOS-Rechner.

- Wie viele Werte (Elemente) sind in den folgenden Vektoren und Matrizen gespeichert?
- Welcher Speicherplatz wird von diesen Vektoren bzw. von diesen Matrizen im Hauptspeicher des Rechners benötigt?

`double m1[10][20][2];`	a) Anzahl Elemente?	b) Speicherbedarf in Bytes?
`int v[1][2][3][4];`	c) Anzahl Elemente?	d) Speicherbedarf in Bytes?
`char my_text[20];`	e) Anzahl Elemente?	f) Speicherbedarf in Bytes?

4.2. Der Anwender gibt die drei Zeichen „**abc**“ mit **`scanf("%19s", my_text);`** in die oben definierte Variable **`my_text`** ein. Direkt nach dem letzten Zeichen wird die Eingabetaste gedrückt. Welcher Wert steht anschließend im Element **`my_text[3]`**?

4.3. Warum ist es eine schlechte Idee, eine Zeichenkette mit dem Befehl **`scanf("%s", my_text);`** eingeben zu lassen? (Kurze Begründung)

4.4. Zur Verarbeitung von Kommazahlen gibt es die Datentypen **`float`** und **`double`**. Wann sollte man den Datentyp **`double`** verwenden? Geben Sie ein Beispiel bzw. einen Grund an.

4.5. Wann sollte man anstelle von **`double`** besser den Datentyp **`float`** verwenden? Geben Sie ein Beispiel bzw. einen Grund an.