

Wintersemester 2021/22

Ingenieurinformatik, Teil 1: C-Programmierung

Schriftliche Fernprüfung mit Videoaufsicht

Prüfer: Jäger-Hezel, Küpper, Krug, Ressel, Tasin, Selting, Reichl

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 die Studiengruppe auf alle Lösungsblätter.
Es werden nur handschriftliche Lösungen auf leeren, weißen DIN-A4-Blättern akzeptiert.

Studierende aus einer alten Studien- und Prüfungsordnung, die zur Kombiprüfung „Ingenieurinformatik“ angemeldet sind, müssen beide Teile (C-Programmierung und Numerik/ Matlab) im selben Semester schreiben.

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

Aufgabe 1: (ca. 26 Punkte)

Schreiben Sie ein C-Programm, welches eine Nullstelle x_0 der Funktion $f(x)$ näherungsweise ermittelt. Die Funktion $f(x)$ ist als Unterfunktion zu programmieren:

$$f(x) = \frac{e^{1023 \cdot x} - 1}{e^{511 \cdot x} - 1} - \frac{255}{63}$$

Der Anwender wird zunächst nach zwei x-Werten x_1 und x_2 gefragt, bei denen die Funktionswerte $f(x_1)$ bzw. $f(x_2)$ unterschiedliche Vorzeichen haben. Hier ist ein konkretes Beispiel:

$$x_1 = 0.001 \quad f(x_1) = -1.37649 < 0$$

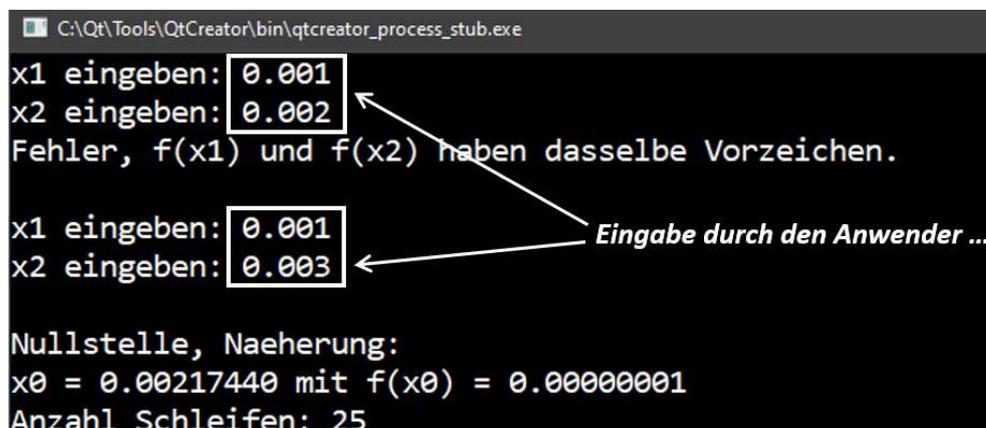
$$x_2 = 0.003 \quad f(x_2) = +1.60218 > 0$$

Zwischen x_1 und x_2 muss sich also (mindestens) eine Nullstelle x_0 befinden. Dort ist $f(x_0) = 0$.

Ihr Programm soll die folgenden Schritte ausführen:

1. Es werden zwei x-Werte x_1 und x_2 über die Tastatur eingegeben.
 2. Nach der Eingabe wird folgendermaßen überprüft, ob die eingegebenen Werte x_1 und x_2 gültig sind:
 - 2.1. Wenn $x_1 \leq 0$ ist oder wenn $x_2 > 2$ ist, dann wird eine Fehlermeldung auf dem Bildschirm ausgegeben und die Eingabe (Schritt 1) wird wiederholt.
 - 2.2. Die Differenz $x_2 - x_1$ muss größer als 10^{-1} sein. Wenn dies nicht der Fall ist, dann wird eine Fehlermeldung ausgegeben und die Eingabe (Schritt 1) wird wiederholt.
 - 2.3. Wenn die Funktionswerte $f(x_1)$ und $f(x_2)$ dasselbe Vorzeichen haben, dann wird eine Fehlermeldung auf dem Bildschirm ausgegeben und die Eingabe (Schritt 1) wird wiederholt.
 3. Solange $x_2 - x_1 > 10^{-10}$ ist, werden die folgenden Schritte in einer Schleife immer wieder ausgeführt:
 - 3.1. Der Wert $x_0 = \frac{x_1 + x_2}{2}$ wird berechnet.
 - 3.2. Wenn das Produkt der Funktionswerte $f(x_0) \cdot f(x_1) > 0$ ist, dann wird der Wert von x_0 in die Variable x_1 geschrieben, sonst wird der Wert von x_0 in die Variable x_2 geschrieben.
 4. Nach dem Ende der Schleife werden x_0 und $f(x_0)$ mit 8 Nachkommastellen ausgegeben.
 5. Die Anzahl der Schleifendurchläufe (vergl. Schritt 3) wird ebenfalls ausgegeben.
-

Der Programmablauf soll ähnlich wie in der folgenden Abbildung aussehen:



```
C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
x1 eingeben: 0.001
x2 eingeben: 0.002
Fehler, f(x1) und f(x2) haben dasselbe Vorzeichen.
x1 eingeben: 0.001
x2 eingeben: 0.003
Nullstelle, Naeherung:
x0 = 0.00217440 mit f(x0) = 0.00000001
Anzahl Schleifen: 25
```

Aufgabe 2: (ca. 16 Punkte)

Eine globale Matrix ist wie folgt definiert:

```
int m[50][60];
```

- 2.1. Schreiben Sie eine C-Funktion mit dem Namen `fill_m`, welche diese Matrix mit ganzzahligen Zufallszahlen im Bereich von 20 ... 70 füllt (beide Grenzen eingeschlossen).
- 2.2. Entwerfen Sie das Struktogramm für eine Funktion `check_m`. Diese Funktion ermittelt die größte Zahl, die in der globalen Matrix `m` abgespeichert ist. Diese Zahl soll auf dem Bildschirm ausgegeben werden. (Die Rückgabe des Ergebnisses an das Hauptprogramm ist nicht erforderlich.)

Auf dem Bildschirm soll auch ausgegeben werden, wie oft diese Zahl in der Matrix `m` vorkommt. (Es ist ja durchaus möglich, dass dieselbe Zahl mehrfach in der Matrix `m` vorkommt.)

Achtung, in diesem Unterpunkt 2.2 soll kein C-Quelltext geschrieben werden!

Aufgabe 3: (ca. 6 Punkte)

In der Mathematik gibt es die Zahlenfolge der sog. „Dreieckszahlen“: 1, 3, 6, 10, 15, 21, 28, 36, ...

Die n -te Dreieckszahl ist die Summe aller ganzen Zahlen von 1 bis n .

Für die fünfte Dreieckszahl gilt also zum Beispiel: $15 = 1 + 2 + 3 + 4 + 5$

Das folgende C-Programm berechnet die ersten 100 Dreieckszahlen. Im Quelltext befinden sich vier Fehler.

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

```
1  #define N 100
2  #include <stdio.h>
3  void dreieck(void);
4  int v(N);
5
6  int main(void)
7  {
8      int i;
9      dreieck;
10     for(i = 0; i < N; ++i) printf("%7d , v[i]");
11     return 0;
12 }
13
14 void dreieck(void)
15 {
16     int i, j;
17     for(i = 0; i < N; ++i)
18     {
19         v[i] = 1;
20         for(j = 0; j < i; ++j)
21             v[i] += j + 2;
22     }
23 }
```

Aufgabe 4: (ca. 5 Punkte)

Wie lautet die Ausgabe der Funktion `test()`, wenn Sie in dem folgenden Programm den Namen `Zarah` eingeben?

```
#include <stdio.h>
void test(char *x);

int main(void)
{
    char str[100];
    printf("Namen: ");
    scanf("%99s", str);
    test(str);
    return 0;
}

void test(char *x)
{
    int i = 0;
    while(x[i] != 0)
    {
        printf("%c %c\n", x[i], x[i] + 1);
        ++i;
    }
}
```

Ausschnitt aus der ASCII-Tabelle:

60 <	61 =	62 >	63 ?
64 @	65 A	66 B	67 C
68 D	69 E	70 F	71 G
72 H	73 I	74 J	75 K
76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S
84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [
92 \	93]	94 ^	95 _
96 `	97 a	98 b	99 c
100 d	101 e	102 f	103 g
104 h	105 i	106 j	107 k
108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s
116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {
124	125 }	126 ~	127 ¨

Aufgabe 5: (ca. 14 Punkte)

5.1. 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?
- Wie viel Speicher wird von diesen Vektoren bzw. Matrizen im Hauptspeicher des Rechners benötigt?

`float x[4][3][2][1];`

a) Anzahl Elemente?

b) Speicherplatz-Bedarf in Bytes?

`int x[1][2][3][4];`

c) Anzahl Elemente?

d) Speicherplatz-Bedarf in Bytes?

`char name[101];`

e) Anzahl Elemente?

f) Speicherplatz-Bedarf in Bytes?

5.2. Der Anwender gibt die drei Zeichen „{|}“ mittels `scanf("%100s", name);` in die Variable `name` ein (siehe Unterpunkt 5.1). Direkt nach dem letzten Zeichen wird die Eingabetaste gedrückt. Welcher Wert steht anschließend im Element `name[3]`?

5.3. Wie lautet die Ausgabe des folgenden Programms?

```
#include <stdio.h>
int main(void)
{
    double d1;
    int z1 = 11, z2 = 4, z3;
    z3 = z1 % z2; printf("%d\n", z3);
    d1 = z1 / z2; printf("%.1f\n", d1);
    return 0;
}
```

5.4. Unterscheidet sich der Speicherplatz-Bedarf der Variablen `d` und `s`? (Kurze Begründung!)

```
double *d;
char *s;
```

5.5. Ein Student fragt seinen Informatiklehrer nach einem Beispielprogramm mit einer „if-Schleife“. Der Informatiklehrer ärgert sich ...

Erläutern Sie, was an dem Begriff „if-Schleife“ so problematisch ist.