Ingenieurinformatik

C-Programmierung

Name	Vorname	Semestergruppe	Hörsaal

	Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Summe

Beginn Ihres Bachelorstudiums:

- ❒ **vor WS13/14 (Kombiprüfung C + MATLAB) ****
- ❒ **von WS13/14 bis WS15/16 ****
- ❒ **von SS16 bis WS17/18 (Kombiprüfung C + MATLAB)**
- ❒ **ab SS18**

**** Die Prüfung ist nur dann gültig, wenn Sie die Zulassungsvoraussetzung erworben haben (erfolgreiche Teilnahme am Praktikum).**

Aufgabensteller: **Dr. Reichl, Dr. Küpper und Kollegen**

Bearbeitungszeit: **60 Minuten**

Hilfsmittel: **Taschenrechner nicht zugelassen,**
PC/Notebook nicht zugelassen,
sonstige eigene Hilfsmittel sind erlaubt,
Bearbeitung mit Bleistift ist erlaubt.

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

Aufgabe 1: (ca. 23 Punkte)

Schreiben Sie ein C-Programm, welches die Geschwindigkeit einer Saturn-V-Rakete während der Brenndauer der ersten Raketenstufe in Zeitschritten der Größe $\Delta t = 0{,}05\ \mathrm{s}$ näherungsweise berechnet. Die Ergebnisse sollen tabellarisch auf dem Bildschirm ausgegeben werden.

Folgende Daten sind gegeben:

- Startmasse der Rakete (inkl. Treibstoff und Nutzlast): $m = 2{,}75 \cdot 10^6\ \mathrm{kg}$
  (Den größten Teil der Startmasse, nämlich 72,7%, macht der Treibstoff der ersten Stufe aus!)
- Treibstoffverbrauch pro Sekunde („Brennrate"): $R = 13{,}5 \cdot 10^3\ \mathrm{kg/s}$
- Schubkraft der ersten Raketenstufe: $F_{\mathrm{Schub}} = 35{,}1 \cdot 10^6\ \mathrm{N}$
- Fallbeschleunigung: $g = 9{,}81\ \mathrm{m/s^2}$

1. Zu Beginn ($t = 0$) befindet sich die Rakete in Ruhe ($v = 0$).
2. Wenn zu einem beliebigen Zeitpunkt t die Masse $m(t)$ und die Geschwindigkeit $v(t)$ bekannt sind, können $m(t + \Delta t)$ und $v(t + \Delta t)$ zum Zeitpunkt $t + \Delta t$ wie folgt berechnet werden:

   $$v(t + \Delta t) = v(t) + \Delta t \cdot a(t) \quad \text{mit der Beschleunigung} \quad a(t) = \frac{F_{\mathrm{Schub}}}{m(t)} - g$$

   $$m(t + \Delta t) = m(t) - R \cdot \Delta t$$

3. Nach jedem Zeitschritt $\Delta t = 0{,}05\ \mathrm{s}$ werden die aktuellen Werte von t (zwei Nachkommastellen) und v (drei Nachkommastellen) tabellarisch ausgegeben.
4. Die Berechnungsschleife wird beendet, wenn der Treibstoff der ersten Stufe zu mehr als 95% aufgebraucht ist.
5. Unmittelbar vor dem Programmende werden die folgenden drei Werte mit drei Nachkommastellen ausgegeben:
   - Die Beschleunigung a beim Start der ersten Raketenstufe (zum Zeitpunkt $t = 0$),
   - die Beschleunigung a unmittelbar vor dem Ende der Berechnungsschleife,
   - der Zeitpunkt t, wann die Rakete erstmals die Schallgeschwindigkeit von $340\ \mathrm{m/s}$ überschreitet.

Die Ausgabe Ihres C-Programms soll so aussehen:

```
C:\Qt\Tools\QtCreator\bin\q...
t =    0.05 s, v =    0.148 m/s
t =    0.10 s, v =    0.296 m/s
t =    0.15 s, v =    0.444 m/s
t =    0.20 s, v =    0.592 m/s
t =    0.25 s, v =    0.740 m/s
t =  140.55 s, v =  1666.909 m/s
t = 140.60 s, v = 1666.909 m/s
t = 140.65 s, v = 1668.479 m/s
t = 140.70 s, v = 1670.050 m/s

Anfangsbeschleunigung: 2.954 m/s^2
Endbeschleunigung: 31.425 m/s^2
Schallgeschwindigkeit bei t = 62.600 s
```

Aufgabe 2: (ca. 22 Punkte)

Das folgende C-Programm prüft, ob ein Vektor sortiert ist (also ob die Werte im Vektor monoton steigend oder monoton fallend sind) oder nicht.

```
#include <stdio.h>
#include <stdlib.h>

#define DIM 5
double vect[DIM];
void fill_random(void);
int test_monoton(void);

int main(void)
{
    int test;
    fill_random();
    test = test_monoton();

    switch(test)
    {
        case            : printf("Monoton steigend\n");
        case            : printf("Monoton fallend\n" );
        case            : printf("Nicht sortiert\n"  );
    }
    return 0;
}

int test_monoton(void)
{
    int idx = 1, steigt = 0, faellt = 0, ret = 10;
    double vorher = vect[0];
    while(idx < DIM)
    {
        if(vect[idx] >= vorher) steigt++;
        if(vect[idx] <= vorher) faellt++;
        vorher = vect[idx++];
    }

    if(steigt == DIM - 1)
        ret = 11;
    else if(faellt == DIM - 1)
        ret = 12;
    return ret;    //// Welchen Wert hat 'idx' in dieser Zeile?
}

void fill_random(void)
{









}
```

2.1. Vervollständigen Sie die switch-case-Anweisung im Hauptprogramm main().

2.2. Schreiben Sie die Definition der Funktion fill_random(). Diese Funktion soll mithilfe einer Schleife alle Elemente des Vektors „vect“ mit unterschiedlichen double-Zufallszahlen im Bereich von 1,5 … 2,5 (beide Grenzen eingeschlossen!) füllen.

2.3. Welchen Wert hat die Variabe „idx“, wenn das Programm in der mit //// markierten Zeile angekommen ist?

idx = ____

2.4. Zeichnen Sie ein Struktogramm, das den genauen Ablauf der Funktion test_monoton() zeigt.

Aufgabe 3: (ca. 10 Punkte)

3.1. Das folgende C-Programm soll ermitteln, wie oft der Buchstabe 'x' in einer Zeichenkette mit einer Länge von max. 50 Zeichen vorkommt. Korrigieren Sie die fünf Fehler im Quelltext!

```
#include <stdio.h>
int count_x(char *str);

int main(void)
{
    int erg;
    char str1[50];
    printf("Text eingeben: ");
    scanf("%50s", str1);
    erg = count_x(str1);
    printf("%d kleine x gefunden!\n, erg");
    return 0;
}

// Wie oft kommt ein kleines x im Text vor?
int count_x(char *str)
{
    int count = 0;
    while (*str != '0')
    {
        if (*str = 'x') count++;
        ++str;

        return count;
    }
}
```

3.2. Die fünf selbst programmierten Funktionen bildschirm_loeschen(), text_ausgeben(), potenzieren(), vektor_fuellen() und mitternachtsformel() werden folgendermaßen aufgerufen:

```
bildschirm_loeschen();

char str[] = "Viel Erfolg bei der Klausur!";
text_ausgeben(str);

double basis = 10.0, ergebnis;
ergebnis = potenzieren(basis, 0.5);

#define ANZAHL 100
float my_vekt[ANZAHL];
vektor_fuellen(my_vekt, ANZAHL);

double a = 1.0, b = 0.0, c = -2.0, x1, x2;
int anzahl_nullstellen;
anzahl_nullstellen = mitternachtsformel(a, b, c, &x1, &x2);
```

Wie sehen die Deklarationen (Prototypen) der 5 Funktionen aus (mehrere Varianten möglich)?

Aufgabe 4: (ca. 12 Punkte)

4.1. Wandeln Sie die folgende Dezimalzahl ins Dualsystem (Binärsystem) um: $245_{DEZ} = ???_{BIN}$
Notieren Sie alle Zwischenschritte Ihrer Berechnung!

4.2. Wandeln Sie dieselbe Dezimalzahl auch ins Hexadezimalsystem um: $245_{DEZ} = ???_{HEX}$

4.3. In einer Speicherzelle im Arbeitsspeicher eines Computers ist die folgende 8-Bit-Binärzahl gespeichert: 0111 1111

Zu dieser Zahl wird 1 addiert.

- Wie lautet das Ergebnis der Addition als Dezimalzahl, wenn es sich bei der 8-Bit-Binärzahl um eine ganze Zahl ohne Vorzeichen handelt?

- Wie lautet das Ergebnis der Addition als Dezimalzahl, wenn es sich bei der 8-Bit-Binärzahl um eine ganze Zahl mit Vorzeichen (also: 8-Bit-Zweierkomplementdarstellung) handelt?

4.4. In einem C-Programm ist eine Matrix wie folgt definiert:

```
double mat[10][10][2];
```

Wie viele Speicherzellen (mit jeweils 8 Bits pro Speicherzelle) werden benötigt, um diese Matrix im Arbeitsspeicher eines Computers abzuspeichern.

(Platz für Notizen und Nebenrechnungen)