Einführung in die Programmierung mit C
Klausur am 25.02.2019

Prof. Dr. Ute Bauer-Wersing
Frankfurt University of Applied Sciences, Fb 2

[redacted]

Klausurkennung

[redacted]

Aufgabe	1	2	3	Σ
Punkte	20	20	30	70
Erreicht				

Allgemeine Hinweise

- Erlaubte Hilfsmittel sind (**in elektronische Version im Vorlagenverzeichnis**):
  - Bauer-Wersing, Behl, Konrad: *Einführung in die Programmierung mit C.*
  - ASCII-Tabelle, Datei `Diverses.h`
- Bei der Benutzung unerlaubter Hilfsmittel oder Kontaktaufnahme mit anderen Klausurteilnehmern gilt die Klausur **sofort** als nicht bestanden.
- Die Klausur besteht aus drei unabhängigen Aufgaben. Lesen Sie die **Aufgabenstellungen sorgfältig**.
- Die Bearbeitungszeit beträgt **120 Minuten**.

Arbeitsanweisungen

- Melden Sie sich unter Ihrer **Klausurkennung** an!
- Entwickeln Sie Ihre Programme in **Ihrem Arbeitsverzeichnis**. Im Unterverzeichnis `vorlagen` Ihres Arbeitsverzeichnisses finden Sie verschiedene Dateien, die Sie benutzen dürfen.
- Nennen Sie die zu erstellenden **Quelldateien** `aufgabe1.c`, `aufgabe2.c` und `aufgabe3.c`!
- **Lösungen**, die bewertet werden sollen, sind im **Unterverzeichnis** `ergebnisse` abzulegen!
- Bitte in keinem Fall mehrere Lösungen für eine Aufgabe abgeben! Sind im Unterverzeichnis `ergebnisse` mehrere Lösungen für eine Aufgabe zu finden, wird die Aufgabe mit 0 Punkten bewertet.
- Schreiben Sie in jedes Programm als Kommentar Ihren Namen und Ihre Matrikelnummer!

Aufgabe 1 (20 Punkte)

Es soll ein Programm entwickelt werden, das mindestens zwei und maximal zehn ganze Zahlen in ein Feld `int feld[10]` einliest und anschließend in der Reihenfolge der Eingabe wieder ausgibt. Dabei soll zwischen zwei aufeinanderfolgenden Zahlen zusätzlich `"<="` ausgegeben werden, wenn die erste der beiden Zahlen kleiner oder gleich der zweiten ist, und `">"` sonst. Im Einzelnen soll das Programm folgendes leisten:

1. Aufforderung (Prompt) des Benutzers, die Anzahl der einzulesenden Zahlen anzugeben.
2. Überprüfung, ob diese Anzahl größer oder gleich zwei und kleiner oder gleich zehn ist. Wenn nicht, soll das Programm mit einer Fehlermeldung abgebrochen werden.
3. Aufforderung (Prompt) des Benutzers, die gewünschte Anzahl von ganzen Zahlen einzugeben, und Einlesen der Zahlen. Ausgabe der Zahlen in der Reihenfolge der Eingabe mit zusätzlich `"<="` oder `">"` zwischen zwei aufeinanderfolgenden Zahlen, wie oben beschrieben.

ösungsbeispiel 1:

```
nzahl der einzugebenden ganzen Zahlen (min 2, max 10): 1
<2, Abbruch.
```

sungsbeispiel 2:

```
zahl der einzugebenden ganzen Zahlen (min 2, max 10): 8
ben Sie 8 ganze Zahlen ein: 1 4 4 3 2 1 4 3
<= 4 <= 4 > 3 > 2 > 1 <= 4 > 3
```

Bitte wenden!

Einführung in die Programmierung mit C
Klausur am 25.02.2019

Prof. Dr. Ute Bauer-Wersing
Frankfurt University of Applied Sciences, Fb 2

Aufgabe 2 (20 Punkte)
Schreiben Sie ein C-Programm, das folgendes leistet:

1. Aufforderung (Prompt) des Benutzers, ein Wort einzugeben, das höchstens 70 Zeichen lang sein darf. Dieses Wort soll in ein Feld `char wort1[71]` eingelesen werden.
2. Aufforderung (Prompt) des Benutzers, ein weiteres Wort einzugeben, das höchstens zehn Zeichen lang sein darf. DieseWort soll in ein Feld `char wort2[11]` eingelesen werden.
3. Ausgabe des ersten Wortes ohne die Zeichen im zweiten Wort.

Lösungsbeispiel

```
Geben Sie wort1 ein (<= 70 Zeichen): Elefantenkuh
Geben Sie wort2 ein (<= 10 Zeichen): Eeh
Wort1 ohne die Zeichen in wort2: lfantnku
```

Aufgabe 3 (30 Punkte)
Ein Zylinder ist durch den Radius seiner Grundfläche `r` und seine Höhe `h` gegeben und in „C" durch nebenstehende Struktur dargestellt. Das Volumen eines Zylinders beträgt $V = \pi r^2 h$.

```
struct cylinder {
     double r,h;
};
```

Schreiben Sie folgende Funktionen:

`int read_cylinder(struct cylinder *arg)`
Die `struct` Variable für einen Zylinder wird per Referenz übergeben. Die Funktion soll Radius und Höhe mit `scanf()` einlesen. Wenn der Radius oder die Höhe negativ eingegeben wurden, soll die Funktion `-1` zurückgeben, sonst `0`.

`double volume_cylinder(struct cylinder *arg)`
Die `struct` Variable für einen Zylinder wird per Referenz übergeben. Die Funktion soll das Volumen des Zylinders berechnen und den Wert zurückgeben. Hinweis: Wenn die Präprozessordirektive `#include <math.h>` benutzt wird, ist die `double`-Konstante `M_PI` mit dem Wert von π definiert.

Schreiben Sie ein Hauptprogramm unter Benutzung der oben beschriebenen Funktionen, das folgendes leistet:

1. Aufforderung des Benutzers, Radius und Höhe von zwei Zylindern einzugeben, und Einlesen der Werte mit der Funktion
2. `int read_cylinder(struct cylinder *arg)`. Dabei soll jeweils anhand des Rückgabewertes überprüft werden, ob die Eingabe korrekt war. Im Fehlerfall soll das Programm mit einer entsprechenden Ausgabe abgebrochen werden.
3. Berechnung der Volumen der beiden Zylindermit der Funktion `double volume_cylinder(struct cylinder *arg)`.
4. Ausgabe von Radius, Höhe und Volumen der Zylinder. Außerdem soll angegeben werden, welcher Zylinder das größere Volumen hat, oder ob ihre Volumen gleich sind.

Lösungsbeispiel 1:

```
Radius und Hoehe des ersten Zylinders: 1.5 3.0625
Radius und Hoehe des zweiten Zylinders: 1.75 2.25
Zylinder 1: Radius 1.500000, Hoehe 3.062500, Volumen 21.647537
Zylinder 2: Radius 1.750000, Hoehe 2.250000, Volumen 21.647537
Die Volumen der Zylinder sind gleich.
```

Lösungsbeispiel 2:

```
Radius und Hoehe des ersten Zylinders: 1.5 4.5
Radius und Hoehe des zweiten Zylinders: 1.5 -3.7
Eingabefehler - Abbruch.
```

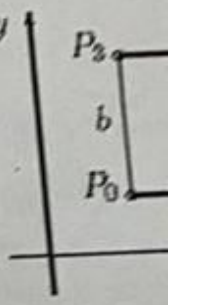
1. Aufforderung (Prompt) des Benutzers, ein Wort einzugeben, das höchstens 70 Zeichen lang sein darf. ein Feld `char wort[71]` eingelesen werden.
2. Das eingelesene Wort soll ausgegeben werden.
3. In der nächsten Zeile soll das Wort noch einmal ausgegeben werden, wobei aber hinter dem letzten von direkt aufeinanderfolgenden Vokalen ein Leerzeichen stehen soll.

Lösungsbeispiel:

```
Geben Sie ein Wort ein (<= 70 Zeichen): Ostereiersucherei
Ostereiersucherei
O ste reie rsu che rei
```

Aufgabe 3 (30 Punkte)

Gegeben seien die Koordinaten eines Punktes P_0 in der xy-Ebene, sowie die Längen a und b. Dann können die x- und y-Koordinaten der Punkte P_1, P_2, P_3 eines wie nebenstehend abgebildeten Rechtecks berechnet werden:



$x_1 = x_0 + a$, $y_1 = y_0$, $x_2 = x_0 + a$, $y_2 = y_0 + b$, $x_3 = x_0$, $y_3 = y_0 + b$

Es soll ein C-Programm geschrieben werden, mit dem die x- und y-Koordinaten eines Punktes, sowie zwei Längen a und b eingelesen werden und die übrigen drei Punkte des entsprechenden Rechtecks berechnet und ausgegeben werden.

Definieren Sie eine C-Struktur, die ein Rechteck darstellt!

```
struct rechteck {
        double x_0, y_0;
        double a, b;
};
```

Die Struktur soll den Eckpunkt $P_0 = (x_0, y_0)$ des Rechtecks, sowie die Seitenlängen a und b enthalten.

Schreiben Sie eine Routine `void print_rechteck(struct rechteck *arg)`, der eine V Rechteckstruktur per Referenz übergeben wird. Die Funktion soll aus den Koordinaten des ersten Pu übrigen Punkte berechnen und anschießend die vier Eckpunkte des Rechtecks und seine Seitenlänge

Ein Hauptprogramm soll folgende Schritte unter Benutzung der oben beschriebenen Funktion ausfüh

1. Aufforderung des Benutzers (Prompt), die x- und y-Koordinaten eines Punktes, sowie die S und b einzugeben.
2. Einlesen der Koordinaten und der Seitenlängen. Wenn die Seitenlängen nicht größer Null einer Fehlermeldung abgebrochen werden.
3. Für den Fall der korrekten Eingabe, müssen die eingelesenen Werte den Punkt-Koordinate Strukturvariablen zugewiesen worden sein.
4. Die vier Punkte des Rechtecks und die Seitenlängen sollen mit der Routine `print_rec`

Lösungsbeispiel 1:

```
Geben Sie x0, y0, a und b ein: 1.5 2.5 5.5 3.5
P0: (1.500000,2.500000)
P1: (7.000000,2.500000)
```

Prof. Dr. Ute Bauer-Wers
Frankfurt University of Applied Sciences, F

Lösungsbeispiel 3:

```
Radius und Hoehe des ersten Zylinders: 1.5 2.5
Radius und Hoehe des zweiten Zylinders: 2.5 1.5
Zylinder 1: Radius 1.500000, Hoehe 2.500000, Volumen 17.671459
Zylinder 2: Radius 2.500000, Hoehe 1.500000, Volumen 29.452431

Zylinder 2 hat das groessere Volumen.
```

Notizen:

3,14 1533