Fakultät für Informatik

Hochschule **Rosenheim**
University of Applied Sciences

Prüfung **WS 2017/18**
Fach: Programmieren 1
Prüfer: Prof. Dr. F.J. Schmitt
Prüfung: 01.02.2018

Bearbeitungszeit:	90 Minuten
Hilfsmittel:	keine

Bearbeiten Sie die 6 Aufgaben ausschließlich auf den Aufgabenblättern (8 Seiten Angaben). Schreiben Sie Ihren vollständigen Namen auf jedes Blatt. Verwenden Sie zum Schreiben keinen Bleistift und keinen Rotstift.
Sollte bei den Angaben eine Teilinformation fehlen oder fehlerhaft sein, so treffen Sie selbst eine geeignete Annahme und dokumentieren Sie sie.

Name: _____ **MatrikelNr.:** _____

Aufgabe 1: Datenstrukturen (15 Punkte)

Nachfolgend soll der Beginn eines Verwaltungsprogramms für einen Getränkehandel gelegt werden.

a) Überlegen Sie sich einen Strukturdatentyp `struct s_produkt`, mit dem Sie ein Getränk verwalten können. Folgende Informationen spielen dabei eine Rolle:
   - Art des Produkts, z.B. “Limonade“, oder “Mineralwasser“
   - Anzahl Lagerbestand des Produkts, z.B. 17, oder 4, oder 0
   - Volumen, als Angabe in Litern, z.B. 0.33, oder 0.75

   Für alle Daten muss in der Struktur tatsächlich Speicherplatz vorgesehen werden. Verwenden Sie für Zeichenketten jeweils eine von Ihnen festzulegende Maximallänge und definieren Sie eine angemessene symbolische Konstante.
   (4 Punkte)

b) Definieren Sie eine Variable `LagerbestandF` als Feld dieser Struktur mit 100 Einträgen.
   (1 Punkte)

c) Schreiben Sie eine Eingabefunktion, mit der ein neuer Datensatz eingelesen werden kann. Die Eingabefunktion wird mit zwei Parametern aufgerufen: einer ganzzahligen Nummer, die die Feldposition bestimmt, sowie den Zielfeldnamen. Ferner soll der Benutzer jeweils angemessen zur Eingabe aufgefordert werden. Die Eingabefunktion hat keinen Rückgabewert.
(5 Punkte)

d) Schreiben Sie eine Funktion, die bestimmt ob ein bestimmtes Getränk vorrätig ist. Die Funktion wird mit drei Parametern aufgerufen: der Feldname, die Anzahl der enthaltenen Datensätze sowie ein String mit dem gesuchten Getränk (z.B. “Limonade“). Die Funktion liefert 1 zurück, falls der Lagerbestand größer 0 ist, ansonsten liefert die Funktion 0 zurück. Anm.: Zu einem Getränk kann es mehrere Einträge mit unterschiedlichem Volumen geben.
(5 Punkte)

Name:

Aufgabe 2: Verständnisfragen zu unterschiedlichen Themen (16 Punkte)

1. Welcher der folgenden Namen ist für Bezeichner nicht zulässig? (1 P)
   a) _Laufvariable1
   b) Variable_a
   c) Gültig_oder_nicht
   d) jahr_2017
   e) gesamt_summe_

   Antwort:

2. Lokale und globale Variablen: Welche Werte haben die (in der jeweiligen Funktion sichtbaren) Variablen i, j jeweils am Ende der Funktion (bei return)? (6P)

```
int i,j;
int main(void)
{
   i = 2;
   i = f() + 2;
   return i;
}

int f(void)
{
   int i = 10;
   for(j = 0; j < 3; j++) i = i + 2;
   g(i);
   return j;
}

int g(int eingabe)
{
   i *= eingabe;
   return i;
}
```

i = ... j = ...

i = ... j = ...

i = ... j = ...

Name:

3. Welche Werte haben die Variablen nach Ausführung des folgenden Code-Fragments? (3P)

```
int a = 10, b = 20, c = 30;
a += 20;
c -= b-- + ++a * 2;
```

Antwort: a = …….. b = …….. c = ……..

4. Gibt es in den nachfolgenden beiden Anweisungen einen inhaltlichen Unterschied? Welche Werte bekommen die Variablen x und y zugewiesen? Begründen Sie Ihre Antwort. (2P)

```
float x, y;
x = 2.0 / 5;
y = 2 / 5;
```

Antwort:

5. Gegeben seien folgende Funktionsdeklaration und Variablendefinitionen:

```
void calculate( int, double );
int n = 5;
int x = 5;
```

Welche zwei Werte werden der Funktion beim Aufruf `calculate( n/2 + 0.1, x+0.5 );` übergeben? (4P)

Antwort:

Aufgabe 3: Zeichenketten (16 Punkte)

Schreiben Sie eine Funktion StringUmdrehen, die als Eingabe einen String erhält und den String umdreht, die Funktion selbst liefert nichts zurück.
Ein Aufruf von StringUmdrehen mit dem String s = “abcd“ führt dazu, dass der String s danach den Wert “dcba“ hat.

Aus der Standardbibliothek darf nur die Funktion strlen() verwendet werden.

Name:

Aufgabe 4: Felder und Reallokation(15 Punkte)

Schreiben Sie eine Funktion `reallokiereArray()`, welcher ein dynamisch allokiertes Array von int-Speicherplätzen und deren Anzahl sowie eine neue Anzahl von Elementen übergeben wird.
Die Funktion erweitert das Array auf die neue Anzahl.
Alle alten Werte bleiben erhalten, neu hinzugekommene Werte werden auf 0 gesetzt.
Die Funktion liefert die Startadresse dieses Arrays und im Fehlerfall NULL zurück.
Sie können davon ausgehen, dass die Anzahl der neu zu allokierenden Speicherplätze größer ist als die bisherige Anzahl von Elementen.

Name:

Aufgabe 5: Listen (26 Punkte)

Stellen Sie sich vor es soll ein Programm entwickelt werden, das dem Anwender ermöglicht einen Projektplan, bestehend aus mehreren aufeinanderfolgenden Knoten zu erstellen. Jeder Knoten soll hierbei eine eindeutige Nummer, einen Namen sowie die Dauer der Aktivitäten enthalten. Durch die Angabe der Dauer einer Aktivität soll die Gesamtlaufzeit des Projekts berechnet werden.

Für die Problemlösung wurden bereits folgende Definitionen festgelegt:

```
#define STRLEN 128

typedef struct s_knoten
{
       int nummer;
       char name[STRLEN];
       double dauer;
       struct s_knoten *next;
}t_knoten;

typedef struct
{
   struct s_knoten *ersterKnoten;
   struct s_knoten *letzterKnoten;
   int anzahlKnoten;
}t_listenkopf;
```

a) Implementieren Sie eine Funktion die bestimmt, ob die Einträge in der Liste nach dem Feld nummer in den Knoten aufsteigend sortiert sind. Ist dies der Fall liefert die Funktion 1 zurück, sonst 0.
Prototyp:
```
int istListeSortiert(t_Listenkopf *li);
```

b) Implementieren Sie eine Funktion, die nach einem bestimmten Knoten sucht. Um einen Knoten suchen zu können, benötigt die Funktion den entsprechenden Anfang der Liste sowie den Namen des Projekts, nach dem gesucht werden soll.
Im Erfolgsfall wird ein Zeiger auf den gefundenen Konten zurückgegeben. Falls es keine Übereinstimmung gibt, soll die Funktion NULL zurückgeben.

Name:

Aufgabe 6: Rekursion (12 Punkte)

a) Welche Ausgabe erzeugt die nachfolgende Funktion baum(), wenn sie mit dem Parameter 3 aufgerufen wird?

```
void baum (int n){
      if (n > 1){
            baum(n-1);
            printf("%d\n", n);
            baum(n-1);
            printf("%d\n", n);
      }
}
```

baum(3) erzeugt folgende Ausgabe:

b) Geben sie die Definition für eine rekursive Funktion `fkt` an, die für zwei übergebene ganze (nicht negative) Zahlen den Funktionswert nach folgender Definition berechnet:

fkt(n, m) = 1, falls m = 0 ist,
= n * fkt(n, m-1) sonst.

c) Was liefert die Funktion aus Teilaufgabe b) bei einem Aufruf fkt(2, 7) zurück?