Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Studiengang ESTB / EEEB

Note / normierte Punkte

Klausur in Programmieren

Sommer 2017, 20. Juli 2017

Dauer: 1,5h

Hilfsmittel: Keine (Wörterbücher sind auf Nachfrage erlaubt)

Name:
MatrikelNr.:

Aufgabe	1	2	3	4	5	6	Summe
Punkte max	12	15	20	14	24	15	100
Punkte							

Alle Fragen beziehen sich auf den Stoff der Vorlesung. Somit sind sie z.B. bezogen auf die Programmiersprache C. Auch sonst gelten die Konventionen wie in unserer Vorlesung.

1. Aufgabe: Grundlagen

Stellen Sie die nachfolgenden Zahlen in den jeweiligen Zahlensystemen dar (12P):

a) 1011 1001 1101 0101 (binär)	hex:	0xB9D5	(1P)
	oktal:	0134725	(3P)
b) 222 (dezimal)	hex:	0xDE	(2P)
	binär:	1101 1110	(2P)
c) C7 (hex)	dezimal:	199	(3P)
	binär:	1100 0111	(1P)

2. Aufgabe: Grundlagen

Schreiben Sie ein vollständiges und lauffähiges C-Hauptprogramm, das die Kantenlängen eines Rechtecks als reelle Zahlen einliest und dessen Fläche berechnet. Die Fläche soll kaufmännisch gerundet werden und als ganze Zahl vorliegen. Negative Werte brauchen nicht berücksichtigt bzw. abgefangen werden. Anschließend soll der gerundete Umfang auf die Konsole ausgegeben werden. Berücksichtigen Sie bei der Ausgabe auf die Konsole eine passende Beschriftung. (15 P)

```
#include <stdio.h>

int main()
{
    float a, b;
    printf(“Kantenlaenge a = “); scanf_s(”%f”, &a);
    printf(“Kantenlaenge b = “); scanf_s(”%f”, &b);

    int iFlaeche = (int)(a * b + 0.5);

    printf(“Gerundeter Flaecheninhalt = %d\n”, iFlaeche);

    return 0;
}
```

3. Aufgabe: Funktionen

a) Erklären Sie aus welchen Teilen eine Funktion besteht (strukturiert vorgehen!) (14 P):

- Funktionskopf, Funktionsrumpf
- Funktionskopf besteht aus
  - Datentyp des Rückgabewertes
  - Funktionsname
  - Parameterliste
    - Parameterliste besteht aus keinem, einem, beliebig vielen Parametern, die durch Komma getrennt werden
    - Parameter besteht aus Datentyp und Parametername

b) Wie bezeichnet man bei folgenden Beispielen die Art, mit der die Parameter der Funktion übergeben werden? Erklären Sie den Mechanismus in **kurzen Stichworten** (6 P):

i. `int function1(double* pdValue)`

call by reference-Aufruf, die Variable wird als Zeiger übergeben und kann geändert werden. Der Zeiger kann auch als Array interpretiert werden.

ii. `void function2(int iValue)`

call by value-Aufruf, nur der Wert wird als Kopie übergeben

- Lösen Sie die Aufgaben bitte auf dem Blatt -

4. Aufgabe: Array/Feld, Indizierung

a) Erklären Sie den Unterschied zwischen einem Array und einer Struktur (**kein Roman – kann Punktabzug geben!**). (4 P)

Bei einem Array besitzen alle Elemente des Arrays denselben Datentyp, bei einer Struktur können sie unterschiedliche Datentypen besitzen. Der Zugriff bei Feldelementen eines Arrays erfolgt über einen Index, bei einer Struktur erfolgt der Zugriff auf die Strukturelemente über den Punktoperator.

b) Schreiben Sie eine kleine Funktion, bei der ein Array mit reellen Messwerten übergeben wird und der Mittelwert bestimmt wird. Der Mittelwert soll als reelle Zahl in Form eines Funktionswertes zurückgegeben werden. Geben Sie anschließend ein Programmfragment an, bei dem die von Ihnen definierte Funktion aufgerufen wird (**d.h. weniger als 5 Anweisungen reichen und bitte kein Hauptprogramm oder irgendwelche Ein- oder Ausgaben schreiben!). Bibliotheken dürfen nicht verwendet werden!** (10 P)

```
float mittelwert(float* afInValues, int iInCount)
{
  float fSumme = 0.0;
  int li;

  for (li = 0; li < iInCount; ++li)
  {
    fSumme += afInValues[li];
  }
  float fMean = iInCount > 0 ? fSumme / iInCount : 0.0f;

  return fMean;
}


  float afValues[3] = {10.1f, 20.2f, 30.3f};
  float fMittelwert;

  fMittelwert = mittelwert(afValues, 3);
```

- Lösen Sie die Aufgaben bitte auf dem Blatt -

5. Aufgabe: Zeichenketten

a) Schreiben Sie eine Funktion strlength, die die Länge einer mit 0 terminierten Zeichenkette bestimmt. Die Zeichenkette soll als Parameter übergeben werden. Der Funktionswert soll die Länge zurückgeben (kein Hauptprogramm, keine Ein- oder Ausgabe!). (10 P)

```
int strlength(char* acInString)
{
  int iIndex = 0;
  while(acInString[iIndex] != 0)
  {
    iIndex++;
  }
  return iIndex;
}
```

b) Schreiben Sie eine Funktion count, die die Anzahl von Zeichen, die KEINE Großbuchstaben sind in einer Zeichenkette acString zählt. (kein Hauptprogramm, keine Ein- oder Ausgabe!). (14 P)

count("This IS An ExamPle!") → 13
Hinweis: Denken Sie daran, dass die Zeichen 'A' … 'Z' wie Zahlen benutzt werden können.

```
unsigned int count(char* acInOutString)
{
  int iIndex = 0;
  unsigned int uiCount = 0;

  while(acInOutString[iIndex] != 0)
  {
    char cCurrent = acInOutString[iIndex];
    if(! ('A' <= cCurrent  &&  cCurrent <= 'Z') )
    {
      ++uiCount;
    }
    ++iIndex;
  }

  return uiCount;
}
```

- Lösen Sie die Aufgaben bitte auf dem Blatt -

6. Aufgabe: Algorithmus

Was machen die nachfolgenden Funktionen unbekannt1-4?

Bitte beschreiben Sie die Funktionsweise möglichst abstrakt – Romane geben Abzug! (15 P)

Hinweis: Testen Sie den Algorithmus anhand eines Funktionsaufrufs und beobachten Sie die Variablenwerte.

```
#include <stdlib.h>

int unbekannt1(int iZahl1, int iZahl2)
{
  int iZahl3 = 1;
  for (int i = 0; i < iZahl2; ++i)
  {
    iZahl3 *= iZahl1;
  }
  return iZahl3;
}

int unbekannt2(double dZahl)
{
  return (int)(dZahl + (dZahl > 0.0? 0.5 : -0.5));
}

double unbekannt3(double dZahl1, int iZahl2)
{
  int iZahl3 = unbekannt1(10, iZahl2);
  return unbekannt2(dZahl1 * iZahl3) / ((double)iZahl3);
}

double* unbekannt4(int iZahl1)
{
  double* adValues = (double*)malloc(iZahl1 * sizeof(double));
  for (int i = 0; i < iZahl1; ++i)
  {
    adValues[i] = unbekannt3(i * 3.1415926539, 3);
  }
  return adValues;
}
```

unbekannt1: Es wird iZahl1 hoch iZahl2 berechnet.
unbekannt2: dZahl wird kaufmännisch gerundet, sowohl für positive als auch negative Zahlen
unbekannt3: dZahl1 wird auf iZahl2-Stellen genau kaufmännisch gerundet
unbekannt4: Es wird ein double-Array der Größe iZahl1 angelegt. Jedes Element wird mit dem Vielfachen von Pi gerundet auf 3 Nachkommastelle initialisiert. Dieses Array wird als Funktionsergebnis zurückgegeben.