



Klausur in Programmieren

Sommer 2009, 16. Juli 2009

Dauer: 1,5h

Hilfsmittel: Keine (Wörterbücher sind auf Nachfrage erlaubt)

Name:

Matrikelnr.:

Aufgabe	1	2	3	4	5	6	Summe
Punkte max	12	6	20	16	26	15	95
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

Welche Ergebnisse haben die folgenden Operationen auf den gegebenen Definitionen? Die Operationen der einzelnen Teilaufgaben bauen nicht aufeinander auf!

```
int a = 5;  
int b = 18;
```

1. `int c = b++;` Variablenwerte: `b = _____`, `c = _____` (2 Pkte)

2. `int c = ++a;` Variablenwerte: `a = _____`, `c = _____` (2 Pkte)

3. `int c = --a / 5;` Variablenwerte: `a = _____`, `c = _____` (2 Pkte)

4. `int c = b % a++;` Variablenwerte: `a = _____`, `b = _____`, `c = _____` (3 Pkte)

5. `int c = ++b % a;` Variablenwerte: `a = _____`, `b = _____`, `c = _____` (3 Pkte)

- Bitte beachten Sie auch die Rückseite -
- Lösen Sie die Aufgaben bitte auf dem Blatt -

2. Aufgabe: Datentypen

Welche Ausgabe produzieren die folgenden Programmfragmente (6P):

a)

```
| short int a = 32767;      // short int    2 Byte
  a++;
  if(a > 0)
  {
      std::cout << "Wert groesser 0. " << endl;
  }
  else
  {
      std::cout << "Wert kleiner oder gleich 0. " << endl;
  }

```

.....

Ausgabe: *Wert*.....*0*.

b)

```
short int b = -32768;     // short int    2 Byte
b++;
if(b > 0)
{
    std::cout << "Wert groesser 0. " << endl;
}
else
{
    std::cout << "Wert kleiner oder gleich 0. " << endl;
}

```

.....

Ausgabe: *Wert*.....*0*.

3. Aufgabe: Array/Feld, Indizierung

a) Für Zeichenketten gibt es unterschiedliche Handhabungen. Die üblichsten Vorgehensweisen bauen auf Arrays mit dem Datentyp `char` auf. Meist füllen die Zeichenketten jedoch nicht das gesamte Array, sodass die Länge oder das Ende einer Zeichenkette gesondert betrachtet werden muss. Es sind dabei zwei Varianten üblich:

1. die Markierung des Endes einer Zeichenkette mit einer 0.
2. die Speicherung der Länge an der 1. Stelle des Arrays.

a) Nennen Sie je Methode einen Vor- und einen Nachteil bei der jeweiligen Vorgehensweise: (4P)

Vorteil bei 1:

Nachteil bei 1:

Vorteil bei 2:

Nachteil bei 2:

b) Schreiben Sie eine Funktion, die eine Zeichenkette der Variante 2 (`acInStringV2`) in eine Zeichenkette der Variante 1 (`acOutStringV1`) umwandelt. Verwenden Sie dabei den nachfolgend vordefinierten Funktionskopf und ergänzen Sie den Funktionsrumpf: (16P)

```
void convertString(char* acOutStringV1, char* acInStringV2)
{
```

```
}
```

4. Aufgabe: Schleifen

a) Welche Schleifentypen gibt es in C/C++? Umreißen Sie in kurzen Stichworten die Unterschiede. (6P)

b) Schreiben Sie nachfolgende while-Schleife als for-Schleife mit derselben Bedeutung (10P):

```
int li = 10;
int lSumme = 0;
while(li >= 0)
{
    std::cout << "Zwischensumme: " << lSumme << endl;
    lSumme += li;
    li--;
}
```

5. Aufgabe: Ganzes Programm

(26 P)

Ergänzen Sie die Lücken im nachfolgenden Programmtext zu einem lauffähigen C++-Programm. Es hat die Aufgabe meteorologische Temperaturangaben einzulesen, daraus den Mittelwert und die Varianz zu bestimmen und auf die Konsole auszugeben. Der Mittelwert und die Varianz bestimmen sich durch

$$mw = \frac{1}{n} \sum_{i=0}^{n-1} x_i, \quad \text{varianz} = \frac{1}{n} \sqrt{\sum_{i=0}^{n-1} (x_i - mw)^2}$$

```
#include <math.h>
#include <iostream>
```

```
using
```

```
// Messwerte von der Konsole einlesen
void eingebenMesswerte(unsigned int& ruiOutMesswerteAnzahl, double*& rpdOutMesswerte)
{
    cout << "Anzahl Messwerte: ";
    cin >> ruiOutMesswerteAnzahl;
    rpdOutMesswerte = new double[ruiOutMesswerteAnzahl];

    for(
        {
            cout << "Messwert: ";
            cin >> rpdOutMesswerte[
        ]];
    }
}
```

```
// Mittelwert bestimmen
double rechneMittelwert(unsigned int uiInMesswerteAnzahl, double* adInMesswerte)
{
    }
}
```

- Bitte beachten Sie auch die Rückseite -
- Lösen Sie die Aufgaben bitte auf dem Blatt -

```

// Varianz bestimmen
double rechneVarianz(double dInMittelwert,
                    unsigned int uiInMesswerteAnzahl, double* adInMesswerte)
{
    double dSumme = 0.0;

    for(              )
    {
        double dDiff = dInMittelwert - adInMesswert[      ];

        dSumme += dDiff *      ;
    }

    return sqrt(dSumme)      ;
}

int main()
{
    double* adMesswerte;
    unsigned int uiMesswerteAnzahl;

    eingabeMesswerte(              );

    double dMittelwert =              ;

    double dVarianz = rechneVarianz(dMittelwert,              );

    cout << "Mittelwert: " <<              ;

    cout << "Varianz: " <<              ;

    delete              ; // Freigabe dynamisches Array

    return 0;
}

```

6. Aufgabe: Algorithmus

Was berechnet die nachfolgende Funktion bzw. was macht sie? Bitte beschreiben Sie die Funktionsweise möglichst abstrakt. (15 P)

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

```
void unknown(int* aiInOutArray, int iInAnzahl)
{
    for(unsigned int uiIndex=0; uiIndex < iInAnzahl; ++uiIndex)
    {
        bool bNoSwap = true;
        for(unsigned int uiIndex2=0; uiIndex2 < uiIndex; ++uiIndex2)
        {
            if(aiInOutArray[uiIndex2] > aiInOutArray[uiIndex2+1])
            {
                int iTmp                = aiInOutArray[uiIndex2];
                aiInOutArray[uiIndex2]  = aiInOutArray[uiIndex2+1];
                aiInOutArray[uiIndex2+1] = iTmp;
                bNoSwap = false;
            }
        }
        if(bNoSwap)
        {
            return;
        }
    }
}
```