

*Dr. K. Simon*

# **Studiengang Maschinenbau und Verfahrenstechnik**

## **1. Vordiplom, Informatik I**

### **Musterlösung**

Herbst 2000

Freitag, 29. September 2000



## Aufgabe 1: C++-Syntax, Typendeklarationen (17 Punkte)

a) i) **Klassifizierung C:** Endlosschleife

**Begründung** In Zeile 7 wird  $i$  nicht erhöht. Die Variable  $i$  ist von Anfang 0 und bleibt 0. Sie wird immer kleiner 2 sein.

iii) **Klassifizierung B:** Fehler, Zeile 7

**Begründung** In Zeile 6 wird die Zeigervariable  $a$  auf 0 gesetzt (d.h. sie zeigt *nirgendwo* hin. In Zeile 7 wird aber auf den Inhalt dieser Zeigervariablen zugegriffen. Das führt zu einem *Laufzeitfehler*.

b) i) 

```
char a;  
char b[100];  
int c;
```

### Varianten

```
int a;  
  
int b[100];  
char *b;  
int *b;
```

ii) **Klassifizierung A:** korrekt

**Begründung**  $a = 4$

iv) **Klassifizierung B:** Fehler, Zeile 7

**Begründung** In Zeile 7 wird auf eine Komponente des Records  $a$  zugegriffen. In Tat und Wahrheit ist aber  $a$  eine Zeigervariable und kein Record. Die korrekte Anweisung müsste lauten:  $a \rightarrow a = 2$ ; Es handelt sich hierbei um einen *syntaktischen Fehler*.

ii) 

```
struct A {  
    bool a;  
    float b;  
};
```

$A\ c;$

```
char a;  
int b;
```

### Varianten

```
struct A {  
    int a;  
    double b;  
};
```

$int\ a;$

$double\ b;$

## Aufgabe 2: Lineare Listen (15 Punkte)

```
a) 1 struct Buchstabe {           // Das Listenelement: ein Buchstabe
2     char inhalt;             // Der Inhalt des Listenelements
3     Buchstabe *Next;        // Der naechste Buchstabe
4 };
5
6 struct Zeichenkette {       // Der Listenkopf: die gesamte Zeichenkette
7     Buchstabe *Start;       // Anfang der Zeichenkette
8     Buchstabe *End;        // Ende der Zeichenkette
9 };

b) 1 void concat (Zeichenkette &s1, Zeichenkette s2)
2     {
3     if (s1.end!=0) {         // Wenn s1 nicht die leere Zeichenkette ist.
4         s1.end->next= s2.start; // verknuepfen wir s1 mit s2
5     } else {                 // ansonsten
6         s1.start= s2.start;   // wird der Anfang von s2 der Anfang des Resultats
7     }
8     s1.end= s2.end;         // das Ende von s2 ist das Ende des Resultats
9 }
```

### Aufgabe 3: Rekursion (21 Punkte)

a)

$$\begin{aligned} \binom{4}{2} &\stackrel{(3)}{=} \binom{3}{1} + \binom{3}{2} \text{ (vorgegeben)} \\ &\stackrel{(3)}{=} \binom{2}{0} + \binom{2}{1} + \binom{3}{2} \\ &\stackrel{(3)}{=} \binom{2}{0} + 2\binom{2}{1} + \binom{2}{2} \\ &\stackrel{(3)}{=} \binom{2}{0} + 2\binom{1}{0} + 2\binom{1}{1} + \binom{2}{2} \\ &\stackrel{(1)}{=} 1 + 2\binom{1}{0} + 2\binom{1}{1} + \binom{2}{2} \\ &\stackrel{(1)}{=} 1 + 2 + 2\binom{1}{1} + \binom{2}{2} \\ &\stackrel{(2)}{=} 1 + 2 + 2 + \binom{2}{2} \\ &\stackrel{(2)}{=} 1 + 2 + 2 + 1 = 6 \end{aligned}$$

b) **Abbruchbedingungen** Gleichungen (1) und (2)

**Rekursionsschritt** Gleichung (3)

```
c) 1  int binomial(int n, int k)
2    {
3      if ((k==0) || (k==n)) {          // Wenn (n 0) oder (n n) berechnet werden soll,
4        return(1);                    // ist das Ergebnis 1 (Gleichungen (1) und (2))
5      } else {                          // ansonsten
6        return(binomial(n-1,k-1)+     // ist das Ergebnis (n-1 k-1)+
7              binomial(n-1,k));       // (n-1 k) (Gleichung (3))
8    }
9 }
```

#### Aufgabe 4: Felder und Records (14 Punkte)

```
1 void MatVecMult(Matrix A, Vektor b, Vektor &x, bool &error)
2 {
3     int i,j;                // Laufvariablen
4
5     error= (A.n!=b.n);      // Wir setzten den Fehler
6     if (!error) {          // Falls kein Fehler passiert ist,
7         x.n= A.m;          // setzen wir die Dimension von x
8         for (i= 0; i<A.m; i++) { // Fuer alle Zeilen von A
9             x.werte[i]= 0.0;    // wird x.werte[i] zuerst auf 0 gesetzt
10            for (j= 0; j<A.n; j++) { // Fuer alle Spalten von A
11                x.werte[i]= x.werte[i]+ // wird zu x.werte[i] das entstpr. Produkt
12                    A.werte[i][j]*b.werte[j]; // von A[i,j] und b[j] addiert.
13            }
14        }
15    }
16 }
```

### Aufgabe 5: Logik (14 Punkte)

a)

$b \leq 2$	$c = 15$	$a \geq 0$	Resultat	Zeile
wahr	wahr	falsch	vielleicht	8
wahr	wahr	wahr	wahr	10
wahr	falsch	wahr	falsch	13
wahr	falsch	falsch	vielleicht	15
falsch	falsch	wahr	falsch	20
falsch	falsch	falsch	vielleicht	22
falsch	wahr	falsch	vielleicht	25
falsch	wahr	wahr	wahr	27

#### b) Beobachtungen

- Immer wenn  $a < 0$ , dann ist das Resultat „vielleicht“.
- Immer wenn  $a \geq 0$  und  $c = 15$ , dann ist das Resultat „wahr“.
- Immer wenn  $a \geq 0$  und  $c \neq 15$ , dann ist das Resultat „falsch“.

#### Darum folgendes vereinfachtes Programm

```
1  int a,b,c;
2
3  ...
4
5  if (a<0) {
6      cout << "vielleicht" << endl;
7  } else if (c==15) {                // (a>=0) && (c==15)
8      cout << "wahr" << endl;
9  } else {                            // (a>=0) && (c!=15)
10     cout << "falsch" << endl;
11 }
```