

# Klausur: Programmieren I

WS06/07

**Erlaubte Hilfsmittel:** keine

Lösung ist auf den Klausurbögen anzufertigen. (eventuell Rückseiten nehmen)

Bitte legen Sie einen Lichtbildausweis und den Studentenausweis auf den Tisch.

**Bearbeitungszeit:** 90 Minuten

Unterschrift

**Benotung**

<b>Aufgabe:</b>	1	2	3	4	5		Gesamt	Note
<b>Punkte:</b>	8	20	24	28	20		100	
<b>erreicht:</b>								

Name:

Matrikelnummer:

---

**Aufgabe 1** Folgende C Programme enthalten Fehler oder Warnungen, die vom Compiler genannt werden. Finden und erklären Sie diese. In jedem Programm ist genau ein solcher Fehler.

```
a) struct Punkt{
2     double x;
3     double y;
4 };
5
6 int main(){
7     Punkt p={17.4,42};
8     return 0;
9 }
```

```
b) #include <stdlib.h>
2 int* f1(int i){
3     int result[i];
4     return result;
5 }
```

Name:

Matrikelnummer:

---

```
c) #include <stdio.h>
2  int main(){
3      int x = 42;
4      int* pX = &x;
5      int* rpX = pX;
6      rpX = (int*)malloc(sizeof(int));
7      rpX = *x;
8
9      return 0;
10 }
```

```
d) int f(int i){
2     return i>0?5:6%5:4*17/5;
3 }
```

Name:

Matrikelnummer:

---

**Aufgabe 2** Rechnen Sie folgende Programme auf dem Papier und geben Sie die Ausgabe auf dem Bildschirm an. Erklären Sie kurz, wie es zu der Ausgabe kommt.

a)

```
#include <stdio.h>
2
3 int main(){
4     printf("%i\n",17/2);
5     printf("%i\n",17%16);
6     printf("%i\n",17/2==8.5?4:17+4*2);
7     return 0;
8 }
```

```
b) #include <stdio.h>
2
3 typedef enum {false,true} bool;
4
5 bool wahr(){
6     printf(" true ");
7     return true;
8 }
9
10 bool falsch(){
11     printf(" false ");
12     return false;
13 }
14
15 bool implies(bool a,bool b){return !a || b;}
16
17 void printbool(bool b){
18     printf("%s\n",b?"true":"false");
19 }
20
21 int main(){
22     printbool(falsch() & falsch());
23     printbool(implies(wahr(),falsch()));
24     printbool(implies(falsch(),wahr()));
25     printbool(!falsch() || falsch());
26     return 0;
27 }
```

Name:

Matrikelnummer:

---

```
c) #include <stdio.h>
2  int y=1;
3
4  int f1(int x){
5      y=y*2;
6      return y*x;
7  }
8
9  int f2(int* x){
10     int y=(*x)*2;
11     return y>(*x);
12 }
13
14 int main(){
15     int y=f1(1);
16     printf("%i\n",y);
17     printf("%i\n",f2(&y));
18     printf("%i\n",f1(1));
19     printf("%i\n",y);
20     return 0;
21 }
```

```
d) #include <stdio.h>
2  typedef enum {A,B,C,D,E} ABC;
3
4  char* f(ABC abc){
5      char* result;
6      switch (abc){
7          case B: printf("B");result="B";
8          case A: printf("A");result="A";
9          case C: printf("C");result="C";
10         case D: printf("D");result="DE";
11         case E: printf("E");result="DE";
12     }
13     return result;
14 }
15
16 int main(){
17     printf("%s\n",f(A));
18     return 0;
19 }
```

Name:

Matrikelnummer:

---

**Aufgabe 3** Gegeben sei folgende Headerdatei für die rekursiv definierten einfach verketteten Listen aus der Vorlesung:

```
1 typedef enum {false,true} bool;
2 typedef void* Object;
3
4 struct PListe{
5     Object head;
6     struct PListe* tail;
7 };
8
9 typedef struct PListe* PList;
10
11 PList nil();
12 PList cons(Object x,PList xs);
13
14 bool isEmpty(PList xs);
15
16 void delete(PList xs);
```

Schreiben Sie Implementierungen für die folgenden Methoden, die der Klasse `PList` zugefügt werden sollen:

- a) `void transform(PList this, Object f(Object))` Es soll jedes Element der Liste mit der übergebenen Funktion transformiert werden. Setzen Sie dazu folgende Gleichungen um:

$$\begin{aligned} \mathit{transform}(\mathit{cons}(x, xs), f) &= \mathit{cons}(f(x), \mathit{transform}(xs, f)) \\ \mathit{transform}(\mathit{nil}(), f) &= \mathit{nil}() \end{aligned}$$

- b) `bool contains(PList this, bool pred(Object))`;

Das Ergebnis soll wahr sein, wenn die Liste ein Element enthält, für das die Anwendung der übergebenen Funktion wahr ergibt. Setzen Sie dazu folgende Gleichungen um:

$$\begin{aligned} \mathit{contains}(\mathit{nil}(), \mathit{pred}) &= \mathit{False} \\ \mathit{contains}(\mathit{Cons}(x, xs), \mathit{pred}) &= \mathit{pred}(x) \vee \mathit{contains}(xs, \mathit{pred}) \end{aligned}$$

Name:

Matrikelnummer:

---

**Aufgabe 4** Schreiben Sie folgende Funktionen:

a) Für Strings ist eine Funktion:

```
int containsUpperCaseChar(char* str)
```

zu implementieren. Sie soll Wahr als Ergebnis liefern, wenn der übergebene String `str` einen Großbuchstaben enthält.

b) Für Strings ist eine Funktion:

```
char* repeat(int i, char c)
```

zu implementieren. Das Ergebnis soll ein String sein, der aus der `i`-fachen Wiederholung des Buchstabens `c` gebildet wird.

Beispielsweise ergibt: `repeat(5, 'A')` den String "AAAAA".

Name:

Matrikelnummer:

---

c) Mit folgender Rekursionsgleichung können Quadratwurzeln angenähert werden:

$$y_{n+1} = \frac{y_n^2 + x}{2 * y_n}$$

Realisieren Sie eine Funktion zur Berechnung der Quadratwurzel, die obige Rekursionsgleichung umsetzt:

```
int quadratwurzel(double yn,double x);
```

Es soll dabei die obige Rekursionsgleichung so oft angewendet werden, bis der absolute Abstand von  $(y_n * y_n)$  zu  $x$  kleiner als  $0.000001$  ist.

Zur Berechnung der Absolutzahl können Sie folgende Funktion benutzen.

```
1 double ab(double d){return d<0?-d:d;}
```

Name:

Matrikelnummer:

---

**Aufgabe 5** Entwerfen Sie in dieser Aufgabe eine kleine Bibliothek zur Darstellung und Berechnung von Vektoren im dreidimensionalen Raum. Ein solcher Vektor bestehe aus drei Fließkommazahlen, für die drei Dimensionen des Raumes je eine.

a) Schreiben Sie eine geeignete Struktur `struct Vektor` für die Bibliothek.

b) Schreiben Sie eine geeignete Konstruktorfunktion, mit der ein Zeiger auf ein Vektordatenobjekt erzeugt und dieses Objekt initialisiert wird.

Name:

Matrikelnummer:

---

- c) Schreiben Sie eine Funktion `greater`, die zwei Vektordatenobjekte als Parameter erhält und einen bool'schen Wert zurückgibt, der anzeigt, ob der Betrag des ersten Objekts größer ist, als der des zweiten.

Der Betrag eines Vektors sei dabei definiert durch:

$$|\vec{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2}$$

- d) Schreiben Sie eine Funktion `move`, die zwei Vektordatenobjekten als Parameter erhält und das erste Objekt um die Werte des zweiten Objekts verschiebt; also das erste Objekt damit verändert.