

# 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, die vom Compiler entdeckt werden. Finden und erklären Sie den Fehler.

```
a) int f1(int y) {  
  2     int f2(int x){  
  3         return 2*x;  
  4     }  
  5  
  6     return f2(y)+2;  
  7 }
```

```
b) int f1(int i){  
  2     static int x=1;  
  3     x=x+i;  
  4     return x;  
  5 }  
  6  
  7 int f2(int y){  
  8     return x*y;  
  9 }
```

Name:

Matrikelnummer:

---

```
c) int f(int i){  
  2   int result=1;  
  3   for (i>0;i--){  
  4       result=result*i;  
  5   }  
  6 }
```

```
d) void f1(int x){  
  2   int y=1;  
  3   y=y+ 2*x;  
  4 }  
  5  
  6 int f2(int y){  
  7   return 2*f1(y);  
  8 }
```

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%4);
6     printf("%i\n",93%51);
7     printf("%i\n",17/2==8.5?17:4);
8     return 0;
9 }
```

```
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 void printbool(bool b){
16     printf("%s\n",b?"true":"false");
17 }
18
19 int main(){
20     printbool(!falsch());
21     printbool(wahr() && falsch());
22     printbool(wahr() || falsch());
23     printbool(wahr() | falsch());
24     printbool(!(wahr() || falsch()));
25     return 0;
26 }
```

Name:

Matrikelnummer:

---

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

Name:

Matrikelnummer:

---

```
d) #include <stdio.h>
2  int main(){
3      int x=0;
4      int y=3;
5      for (;y>0;y--){
6          while (x<5){
7              x=x+1;
8              if (x%2==0) continue;
9              printf("x: %i y: %i\n",x,y);
10         }
11     }
12     return 0;
13 }
```

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) `Object ithElement(PList this, int i)` Das Ergebnis soll das  $i$ -te Element der Liste zurückgeben. Setzen Sie dazu folgende Gleichungen um:

$$\begin{aligned} \text{ithElement}(\text{Cons}(x, xs), 0) &= x \\ \text{ithElement}(\text{Cons}(x, xs), i) &= \text{ithElement}(xs, i - 1) \end{aligned}$$

- b) `bool isOrdered(PList this, bool smallerThan(Object, Object));`

Das Ergebnis soll wahr sein, wenn die Liste sortiert ist, wobei der Funktionsparameter eine Funktion ist, die `true` als Ergebnis liefert, wenn das erste Argument kleiner als das zweite ist. Setzen Sie dazu folgende Gleichungen um:

$$\begin{aligned} \text{isOrdered}(\text{Empty}(), \text{smallerThan}) &= \text{True} \\ \text{isOrdered}(\text{Cons}(x, \text{Empty}()), \text{smallerThan}) &= \text{True} \\ \text{isOrdered}(\text{Cons}(x, \text{Cons}(y, ys)), \text{smallerThan}) \\ &= \text{smallerThan}(x, y) \wedge \text{isOrdered}(\text{Cons}(y, ys), \text{smallerThan}) \end{aligned}$$

**Aufgabe 4** Schreiben Sie folgende Funktionen:

a) Für Strings ist eine Funktion:

```
int howOftenContained(char* str, char c)
```

zu implementieren. Es soll zurückgegeben werden, wie oft der Buchstabe `c` in dem String `str` vorkommt.

b) Für Strings ist eine Funktion:

```
int map(char* fromStr, char* toStr, char f(char))
```

zu implementieren. Für jeden Buchstaben des Parameters `fromStr` soll der Funktionsparameter `f` angewendet werden und das Ergebnis dieser Anwendung im Parameter `toStr` an der entsprechenden Position gespeichert werden.

Name:

Matrikelnummer:

---

c) Die folgende Funktion soll realisiert werden:

```
int hoch(double x,unsigned int y);
```

Das Ergebnis sei die Potenz zu  $y$  von  $x$ , also  $hoch(x, y) = x^y$

Schreiben Sie sowohl eine iterative als auch eine rekursive Version dieser Funktion.

Verwenden Sie dabei keine der Funktionen aus den Standardbibliotheken.

Name:

Matrikelnummer:

---

**Aufgabe 5** Entwerfen Sie in dieser Aufgabe eine kleine Bibliothek zur Darstellung und Berechnung von Wetterdaten. Ein Wetterdatensatz besteht dabei aus einem Wert für die Temperatur, einen Wert der Luftfeuchtigkeit und einem Wert des Luftdrucks.

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

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

Name:

Matrikelnummer:

---

- c) Schreiben Sie eine Funktion, die zwei Wetterdatenobjekte als Parameter erhält und einen bool'schen Wert zurückgibt, der anzeigt, ob das erste Wetterdateobjekt eine höhere Temperatur hat als das zweite.

- d) Schreiben Sie eine Funktion, die ein Wetterdatenobjekt als Parameter erhält und für dieses Objekt die Luftfeuchtigkeit halbiert; also das Objekt verändert.