

Probeklausur: Programmieren I

WS09/10

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

Aufgabe:	1	2	3	4	5	6		Gesamt	Note
Punkte:	12	16	18	20	18	16		100	
erreicht:									

Name:

Matrikelnummer:

Aufgabe 1 Führen Sie folgendes Programm von Hand aus und notieren Sie die sich ändernden Werte der einzelnen Variablen.

a) `Aufgabe1a.c`

```
1 #include <stdio.h>
2
3 int main(){
4     int x=0;
5     int y=1;
6     for (x=10;x!=y;x=x-y){
7         if (y>x) break;
8         y=y+1;
9         printf("x=%d y=%d\n",x,y);
10    }
11
12    return 0;
13 }
14
```

Name:

Matrikelnummer:

b) `Aufgabe1b.c`

```
1 #include <stdio.h>
2
3 int main(){
4     int x=52;
5     switch (x%4==0){
6         case 1: x=x+1;
7         case 2: x=x+2;
8         default: x=x+3;
9     }
10    printf("x=%d\n", x);
11    return 0;
12 }
```

Name:

Matrikelnummer:

c) `Aufgabe1c.c`

```
1 #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 }
```

Aufgabeld.c

d)

```
1 #include <stdio.h>
2
3 typedef struct {int x;int y;} Punkt;
4
5 Punkt verschiebe(Punkt p,int dX,int dY){
6     Punkt result=p;
7     p.x=p.x+dX;
8     p.y=p.y+dY;
9     return result;
10 }
11
12 void printPunkt(Punkt p){
13     printf("( %i, %i)\n",p.x,p.y);
14 }
15
16 int main(){
17     Punkt p={17,4};
18     printPunkt(p);
19     Punkt p2 = verschiebe(p,25,38);
20     printPunkt(p);
21     printPunkt(p2);
22     return 0;
23 }
```

Aufgabe 2 Schreiben Sie folgende Funktionen auf Zeichenketten:

a) Für Strings ist eine Funktion:

```
boolean isPalindrom(char* str1);
```

zu implementieren. Es soll zurückgegeben werden, ob ein String Palindrom ist. (Palindrom sind Wörter, die vorwärts und rückwärts gelesen gleich sind, z.B.: Otto, Anna, Lagerregal, Rentner)

b) Für Strings ist eine Funktion:

```
unsigned int howOftenContaind(char* str, char c)
```

zu implementieren. Es soll berechnet werden, wie oft das Zeichen `c` im String `str` enthalten ist.

Name:

Matrikelnummer:

Aufgabe 3 Realisieren Sie die folgenden mathematischen Funktionen in C

a)

$$p(n) = \begin{cases} 0, & \text{falls } n = 0, \\ n - 1, & \text{sonst} \end{cases}$$

b)

$$s(m, n) = \begin{cases} m, & \text{falls } n = 0, \\ p(s(m, n - 1)), & \text{sonst} \end{cases}$$

Name:

Matrikelnummer:

c)

$$f(m, n) = \begin{cases} 0 & \text{falls } m = 0 \\ f(m - 1, 1) & \text{falls } m > 0, n = 0 \\ f(m - 1, f(m, n - 1)) & \text{falls } m > 0, n > 0 \end{cases}$$

Name:

Matrikelnummer:

Aufgabe 4 Entwerfen Sie in dieser Aufgabe eine kleine Bibliothek zur Darstellung von Studentendaten. Ein Studentendatensatz besteht dabei aus einem Wert für Vorname, einem Wert für den Nachname, einen Wert für das Fachsemester und einen Wert für die Anzahl der bereits erreichten Creditpoints.

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

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

Name:

Matrikelnummer:

- c) Schreiben Sie eine Funktion, die zwei Studentenobjekte als Parameter erhält und einen bool'schen Wert zurückgibt, der anzeigt ob der erste Student in einem niedrigeren Fachsemester ist, und mehr Creditpoints erreicht hat.

- d) Schreiben Sie eine Funktion, die als ersten Parameter ein Studentenobjekt erhöht und als zweiten Parameter ein Integer. Der zweite Parameter gibt die Anzahl der Creditpoints an, die dem Student gut geschrieben werden sollen. Hinweis Das Objekt wird verändert!

Name:

Matrikelnummer:

Aufgabe 5 Gegeben Sei folgende Implementierung von einfach verketteten Listen, wie aus der Vorlesung bekannt. Die Listenelemente sind ganze Zahlen

```
1 typedef enum {false,true} boolean;
2
3 struct Liste{
4     boolean isEmpty;
5     int head;
6     struct Liste* tail;
7 };
8
9 typedef struct Liste Li;
10 Li* newLi(char head,Li* tail);
11 Li* newEmptyLi();
```

Implementieren Sie folgende Funktionen rekursiv:

a) `int sum(Li* this);`

Es soll die Summe aller in der Liste gespeicherten Elemente berechnet werden. Für die leere Liste sei das Ergebnis 0.

Name:

Matrikelnummer:

b) `int last(Li* this);`

Es soll da letzte Listenelement zurück gegeben werden. Für leere Listen soll auch hier die 0 als Ergebnis zurückgegeben werden.

Name:

Matrikelnummer:

Aufgabe 6 Erklären Sie in kurzen Worten.

- a) Strings werden in C als Zeiger auf das erste Zeichen der Zeichenkette dargestellt. Auch Arrays sind Zeiger auf das erste Element der Reihe.

Trotzdem gibt es einen entscheidenden Unterschied zwischen

```
char* str = (char*)malloc(...)
```

und

```
char str [] = {...}
```

- b) In C-Funktionen können statische Variable deklariert werden. Was bedeutet das?

Name:

Matrikelnummer:

- c) Welche Möglichkeit hat man, wenn man eine Reihung als Zeiger auf das erste Element übergeben bekommt, die Anzahl der Elemente in dieser Reihung zu bestimmen?

- d) Was ist bei folgenden Code problematisch?

```
1 int* f(int i){
2     int result = 42*i;
3     printf("%d\n", result);
4     return &result;
5 }
```