Name, Vorname, Unterschrift:	Matrikelnummer:

Klausur
Kompaktkurs Einführung in die Programmierung
Dr. S. Zimmer & M. Sedlacek
19. März 2010

Wichtig:

- Die Angabe besteht aus 6 Seiten — prüfen Sie bitte Ihr Exemplar auf Vollständigkeit.
- Kennzeichen Sie jedes Blatt lesbar mit Ihrem Namen und dieses Deckblatt zusätzlich mit Ihrer **Matrikelnummer und Unterschrift**; trennen Sie die Heftung nicht auf.
- Bearbeiten Sie die Aufgaben auf den Angabenblättern. Sollte der Platz für einen Aufgabenteil nicht reichen, benutzen Sie bitte die Rückseite der Angabenblätter und machen Sie einen entsprechenden Vermerk bei der Aufgabe.
- Insgesamt können Sie 40 Punkte erreichen, bei jeder Aufgabe sind die für die jeweiligen Teilaufgaben vorgesehenen Punkte angegeben. Zum Bestehen sind 17 Punkte erforderlich.
- Als Hilfsmittel ist nur ein Ausdruck der Vorlesungsfolien erlaubt (handschriftliche Anmerkungen sind OK).
- Eigene Funktionen dürfen in den folgenden Teilaufgaben weiterverwendet werden; von den Standardfunktionen sind, soweit nicht bei der Aufgabe anders angegeben, nur die zur Ausgabe zulässig (`std::cout` bzw. `printf`, was immer Ihnen davon lieber ist).
- Die Bearbeitungszeit beträgt 75 Minuten.

Aufg.	1a)	1b)	1c)	1d)	2a)	2b)	2c)	2d)	3a)	3b)	4a)	4b)
Pkt.	/2	/2	/2	/2	/3	/2	/4	/4	/3	/7	/3	/6

Summe	/40
Note	

1 Ausdrücke und Zeiger (max. 2+2+2+2=8 Punkte)

Geben Sie für folgende Programmausschnitte jeweils den Inhalt der geforderten Variablen nach Ablauf an.

a) Was ist der Inhalt von `a`, `b` und `c`?

```
int a = 0, b = 1, c = 2;
a = b;
b = c;
c = a;
```

b) Was ist der Inhalt von `a` und `b`?

```
int a = 0, b = 1;
b -= ++a;
```

c) Was ist der Inhalt von `a` und `b`?
`0` ist dabei die Ziffer Null und nicht der Buchstabe O.

```
char a = '0', b = '0';
if (a-b) a++;
if (b) b++;
```

d) Was ist der Inhalt von `a` und `b`?

```
int a = 5, b = 2;
int *p = &b;
a = *p + 4;
p = &a;
b = 3 + *p - 1;
```

2 Funktionen auf Zeichenketten (max. 3+2+4+4=13 Punkte)

Im Folgenden soll eine Reihe von Funktionen realisiert werden die auf Zeichenketten operieren und unterschiedliche Rückgabewerte haben. Dabei haben alle Parameter den Typ `char *`.

a) Schreiben Sie eine Funktion mit einer Zeichenkette als Parameter, die die Länge dieser übergebenen Zeichenkette ohne das abschließende `'\0'` zurückgibt.

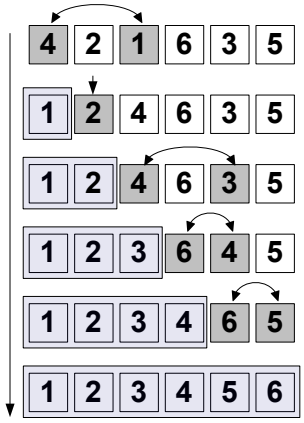
b) Schreiben Sie eine Funktion die eine Zeichenkette zurückgibt, die durch Entfernen des ersten Zeichens einer übergebenen Zeichenkette entsteht. Hierbei soll kein neuer Speicher angelegt werden. Sie können davon ausgehen, dass die übergebene Zeichenkette mindestens die Länge `1` hat.
Beispiel:
`cutHead('Hallo')` liefert `'allo'`

c) Schreiben Sie nun eine Funktion `bool isBeginning(char* teilstr, char* str)` die testet, ob eine Zeichenkette am Anfang einer zweiten Zeichenkette steht. Dabei soll `true` zurückgegeben werden wenn die Zeichenkette `teilstr` am Anfang der Zeichenkette `str` steht, andernfalls `false`. Dabei kann vorausgesetzt werden, dass `teilstr` höchstens gleich lang wie `str` ist.
Beispiele:
`isBeginning('tum','tumult')` liefert `true`
`isBeginning('TUM','tumult')` liefert `false`
`isBeginning('mul','tumult')` liefert `false`

d) Schreiben Sie nun eine Funktion `bool isPart(char* teilstr, char* str)`, die prüft ob eine Zeichenkette irgendwo in einer anderen Zeichenkette vorkommt. Dabei soll `true` zurückgegeben werden falls `teilstr` irgendwo in `str` enthalten ist, andernfalls `false`. Hier darf nicht mehr vorausgesetzt werden, das die Länge von `teilstr` kleiner als die von `str` ist.
Beispiele:
`isPart('gramo','programmieren')` liefert `false`
`isPart('gramm','programmieren')` liefert `true`

3 Sortierung mit Selectionsort (max. 3+7=10 Punkte)

Selectionsort ist ein zu Bubblesort ähnlicher Sortieralgorithmus. Im Folgenden soll ein Feld ganzer Zahlen mit diesem Verfahren sortiert werden. Das zu sortierende Feld wird in zwei Teile zerlegt. Im ersten Teil befinden sich die bereits sortierten Elemente und im zweiten Teil die noch unsortierten. In jedem Schritt wird nun ein kleinstes Element des unsortierten Teils bestimmt und an das Ende des bereits sortierten Teils verschoben, indem es mit dem ersten Element des unsortierten Teils getauscht wird. In jedem Schritt vergrößert sich dadurch der sortierte Teil des Feldes. Zu Beginn ist der sortierte Teil leer. Das Verfahren endet, wenn der unsortierte Teil leer ist, also alle Elemente in den sortierten Teil übertragen wurden. Als Beispiel dient die Grafik rechts.



a) Schreiben Sie eine Funktion zum Vertauschen zweier Werte eines Feldes. Dabei soll die Funktion sowohl das Feld übergeben bekommen, in welchem die beiden Elemente vertauscht werden sollen, als auch die beiden Indizes der Elemente. Ein Funktionsergebnis gibt es nicht.

b) Realisieren Sie nun unter Verwendung von Teilaufgabe a) den Selectionsort–Algorithmus. Nehmen Sie dazu an, dass Ihre Funktion ein unsortiertes Feld und dessen Länge übergeben bekommt. Zurückgegeben wird nichts, da auf dem übergebenem Feld operiert wird. Die Signatur könnte damit so aussehen: `void selection_sort(int* feld, int len)`

Name, Vorname:

4 Listen (max. 3+6=9 Punkte)

In dieser Aufgabe sollen einfach verkettete Listen betrachtet werden.

a) Zunächst als abstrakter Datentyp (Sorten und Operationen wie in der Vorlesung definiert). Gegeben sei eine Funktion $f : T \times \mathit{List} \to \mathit{List}$ mittels folgender Identitäten:

$$\begin{aligned} \forall x \in T &: f(x, \mathrm{NewList}) = \mathrm{NewList} \\ \forall x \in T : \forall L \in \mathit{List} &: f(x, \mathrm{Cons}(x, L)) = f(x, L) \\ \forall x, y \in T \text{ mit } x \neq y : \forall L \in \mathit{List} &: f(x, \mathrm{Cons}(y, L)) = \mathrm{Cons}(y, f(x, L)) \end{aligned}$$

Beschreiben Sie, was die Funktion f „tut“ – geben Sie insbesondere das Ergebnis von $f(x, L)$ an für $x = 10$ und L als Liste mit den Komponenten $(10, 20, 10, 30)$.

b) Implementieren Sie die Funktion aus Teilaufgabe a) in der Datenstruktur aus der Übung – T ist `int`, Listenelemente als Strukturtyp

```
struct element {
   int inhalt;
   struct element *naechstes;
};
```

Ihre Funktion `f` soll die als Parameter übergebene Liste nicht verändern, sondern alle Elemente der neuen Liste neu erzeugen.