
Name, Vorname

Matrikelnummer

Probeklausur zur Vorlesung
Einführung in die Programmierung

WS 2008/09

Dauer: 2 Stunden

Hinweise:

- Schreiben Sie Ihren Namen und Ihre Matrikelnummer auf dieses Deckblatt und auf alle Klausurbögen. Diese Aufgabenstellung und alle Klausurbögen müssen am Ende abgegeben werden.
- Alle Antworten sind auf diesem Aufgabenblatt einzutragen, *nicht* auf dem Klausurbogen. Den Klausurbogen können Sie bei Bedarf als Schmierblatt verwenden.
- Die Aufgabenstellung muss mit abgegeben werden, auch wenn die Aufgaben nicht bearbeitet wurden.
- Es sind *keine* Hilfsmittel zugelassen.
- Die Bearbeitung der Aufgaben darf nur mit dokumentenechten Stiften erfolgen, z.B. mit Kugelschreiber oder Füller, *nicht* mit Bleistift. Verwenden Sie keine Stifte, die in rot schreiben.
- Mobiltelefone sind auszuschalten.
- Es ist nur ein Toilettengang erlaubt.

Viel Erfolg!

Bewertung:

Aufgabe	1	2	3	4	5	6	7	8	Summe
Maximalpunkte	6	4	6	6	8	6	8	10	54
erreichte Punkte									

(Note)

(Datum)

(1. Prüfer)

(Datum)

(2. Prüfer)

Aufgabe 1: (Zahlendarstellung)

(6 Punkte)

Rechnen Sie 60 – 70 jeweils im Einer- und Zweierkomplement mit einer Wortlänge von 8 Bit. **Hinweise:** $60 = 2^5 + 2^4 + 2^3 + 2^2$ und $70 = 2^6 + 2^2 + 2^1$

Nebenrechnungen:

Rechnung im Einerkomplement:

+							
						+	

Rechnung im Zweierkomplement:

+							
						+	

Aufgabe 2: (Zahlendarstellung)

(4 Punkte)

(a) Wandeln Sie die folgende Zahl in die hexadezimale Darstellung um.

$(213)_4 =$ _____

(b) Wandeln Sie die folgende Zahl in die oktale Darstellung um.

$(106)_{10} =$ _____

(c) Wandeln Sie die folgende Zahl in die duale Darstellung um.

$(213)_4 =$ _____

(d) Wandeln Sie die folgende Zahl in die binäre Darstellung um.

$(201)_3 =$ _____

Aufgabe 3: (Quickies)

(6 Punkte)

(a) Nennen Sie mindestens fünf Grunddatentypen von C.

(b) Nennen Sie die arithmetischen Operatoren in C.

(c) Kreuzen Sie im Folgenden die richtigen Aussagen an. Das Schlüsselwort `static` bedeutet bei lokalen Variablen innerhalb einer Funktion:

- die Variable ist nach außen nicht sichtbar
- die Variable bleibt nach Beenden der Funktion im Speicher erhalten
- die Variable wird bei jedem Aufruf der Funktion initialisiert
- die Variable wird in einem schnellen CPU-Register abgelegt
- die Variable wird nur beim ersten Aufruf initialisiert

(d) Welche der folgenden benutzerdefinierten Namen sind in C zulässig?

- | | zulässig | unzulässig |
|------------------------|--------------------------|--------------------------|
| (1) <code>--up</code> | <input type="checkbox"/> | <input type="checkbox"/> |
| (2) <code>US\$</code> | <input type="checkbox"/> | <input type="checkbox"/> |
| (3) <code>_main</code> | <input type="checkbox"/> | <input type="checkbox"/> |
| (4) <code>hoch3</code> | <input type="checkbox"/> | <input type="checkbox"/> |

(e) Schreiben Sie einen logischen Ausdruck in C-Notation, der zu drei gegebenen Zahlen a , b und c feststellt, ob a , b und c die Seiten eines Dreiecks sein können.

Aufgabe 4: (C-Programmierung)

(6 Punkte)

(a) `char *str = "Informatik";`
`char c = *(str + 4);`

Welchen Wert hat c nach der Initialisierung? _____

(b) Welcher der folgenden Ausdrücke ist äquivalent zu `c = *(str + 4);`?

- `c = *str[4];`
- `c = str[4];`
- `c = &str[4];`
- `c = str[0] + 4;`

(c) `int a, b, c;`
`a = 4;`
`b = 6;`
`c = (++a) * (b--);`

Welchen Wert hat c nach der Zuweisung? _____

(d) `int b = 5;`
`int *c = &b;`
`c++;`

Welchen Wert hat b? _____

(e) Erklären Sie, warum das folgende Programm nicht korrekt ist.

```
#include <stdio.h>
void main(void) {
    char *str = "Hallo, Welt!";
    str[2] = 'x';
    printf("%s\n", str);
}
```

Aufgabe 5a: (C-Programmierung)

(4 Punkte)

Schreiben Sie eine Funktion `max`, die aus einer Folge von Zahlen (`int`-Werte) die größte Zahl ermittelt (Maximumbestimmung) und zurückliefert. Die Zahlenfolge soll als Parameter an die Funktion übergeben werden. Es dürfen keine globalen Variablen verwendet werden.

Aufgabe 5b: (C-Programmierung)

(4 Punkte)

Schreiben Sie eine rekursive Funktion, die n^{2^k} berechnet. Es dürfen keine globalen Variablen verwendet werden.

$$n^{2^k} = \begin{cases} n & , \text{ falls } k = 0 \\ n^{2^{k-1}} \cdot n^{2^{k-1}} & , \text{ falls } k > 0 \end{cases}$$

Aufgabe 6a: (C-Programmierung)

(3 Punkte)

Betrachten Sie unten stehende C-Funktion `avg`, die den Durchschnittswert von `n` gegebenen `int`-Werten berechnet. Die Zahlen werden als Vektor (Array) an die Funktion übergeben, die Anzahl der Zahlen wird ebenfalls an die Funktion übergeben. Ergänzen Sie die fehlenden Zeichenfolgen.

Hinweis:

$$\text{avg}(a_1, a_2, \dots, a_n) := \frac{a_1 + a_2 + \dots + a_n}{n}$$

```

..... avg(....., int n) {

    long sum = 0;

    int i;

    for (i = 0; i < n; .....)

        sum ..... a[i];

    return .....;
}

```

Aufgabe 6b: (C-Programmierung)

(3 Punkte)

Definieren Sie ein Makro in C, das von 2 gegebenen Rechtecken `x` und `y` das Rechteck mit größtem Flächeninhalt liefert. Ein Rechteck sei definiert durch seine zwei Seiten, daher ist der Rechteck-Datentyp definiert als

```

typedef struct {
    int a, b;
} rechteck_t;

```

Aufgabe 7a: (C-Programmierung) (4 Punkte)

Schreiben Sie eine Funktion `encode`, die einen gegebenen String kodiert, indem das Alphabet um n Stellen verschoben wird (Cäsar-Codierung). Aus dem Wort `Elefant` wird bei $n = 2$ das Wort `Gnghcpv`. Umlaute wie `ä` oder `ü` müssen von Ihrem Programm nicht berücksichtigt werden.

Die Kodierung soll nur erfolgen, falls $1 \leq n \leq 9$ ist, andernfalls wird die Funktion mit Rückgabe eines Fehlerwertes beendet. Es dürfen keine globalen Variablen verwendet werden.

Aufgabe 7b: (C-Programmierung) (4 Punkte)

Die Koeffizienten eines Polynoms sollen in einem Array gespeichert werden. Bspw. können die Koeffizienten des Polynoms $16x^3 - 19x^2 - 27x + 13$ in einem Array `a` wie folgt abgespeichert werden:

i	3	2	1	0
a[i]	16	-19	-27	13

Schreiben Sie eine Funktion, die die mathematische Ableitung eines solchen Polynoms berechnet. Die Ableitung des obigen Polynoms wäre $48x^2 - 38x - 27$. Das Array soll als IN/OUT-Parameter an die Funktion übergeben werden. Es dürfen keine globalen Variablen genutzt werden.

Aufgabe 8: (C-Programmierung)

(10 Punkte)

Ergänzen Sie die im folgenden Programm fehlenden Funktionen `append`, `getFirst` sowie `isEmpty`, so dass ein lauffähiges Programm entsteht. Erklärungen zu den Funktionen:

- `void append(list_t *l, int v)` fügt ein neues Element mit Wert `v` am Ende der Liste `l` ein.
- `int getFirst(list_t *l)` gibt das erste Element der Liste `l` zurück und entfernt es aus der Liste
- `char isEmpty(list_t *l)` zeigt an, ob die Liste `l` leer ist, also keine Elemente speichert

```
#include <stdio.h>
#include <stdlib.h>

typedef struct element {
    int value;
    struct element *next;
} element_t;

typedef struct {
    int size;
    element_t *first;
    element_t *last;
} list_t;

list_t *create(void) {
    list_t *res;

    res = (list_t *) malloc(sizeof(list_t));
    res->first = NULL;
    res->last = NULL;
    res->size = 0;

    return res;
}

int main(void) {
    int i;
    list_t *l = create();

    for (i = 1; i < 20; i++)
        append(l, i);

    while(!isEmpty(l))
        printf("%d\n", getFirst(l));

    return 0;
}
```