Universität Mannheim
Fakultät für Mathematik und Informatik
Lehrstuhl für Praktische Informatik IV
Professor Dr. W. Effelsberg

Teilprüfung
Software- und Internet-Technologie
Oktober 2002: Programmierkurs 2

Name: .... Vorname: ....

Matrikel-Nr.: .... Semester: .... Fach: ....

Hinweise:

1. Bitte füllen Sie sofort den Kopf des Deckblattes aus.
2. Unterschreiben Sie die Klausur auf der letzten Seite.
3. Überprüfen Sie bitte Ihr Klausurexemplar auf Vollständigkeit (**9** Seiten).
4. Tragen Sie die Lösungen – soweit möglich – direkt in die Klausur ein.
5. Zugelassene Hilfsmittel: nicht programmierbarer Taschenrechner
6. Bearbeitungszeit: 66 Minuten.

Aufgabe	max. Punktzahl	Punkte
1	9	
2	9	
3	11	
4	14	
5	11	
6	12	
Summe	66	

Aufgabe 1: Multiple-Choice-Aufgaben *[9 Punkte]*

Beantworten Sie die folgenden Fragen bzgl. der Sprache C (es ist jeweils genau eine Lösung richtig):

a) [1,5 Punkte] Welchen Wert hat `a` nach Ausführung der folgenden Anweisung:

```
int a = 25 & 11;
```

- ☐ 1
- ☐ 9
- ☐ 18
- ☐ 25
- ☐ keinen, da die Operation nicht zulässig ist.

b) [1,5 Punkte] Welchen <u>Dezimal</u>wert hat `a` nach Ausführung der folgenden Anweisung:

```
int a = 0x517;
```

- ☐ 0
- ☐ 335
- ☐ 517
- ☐ 1303
- ☐ keinen, da die Operation nicht zulässig ist.

c) [1,5 Punkte] Welchen Wert hat `c` nach Ausführung der folgenden Anweisungen:

```
char *s = "ein string";
char c = s[10];
```

- ☐ 'g'
- ☐ ' ' *(Leerzeichen)*
- ☐ '\0'
- ☐ '\n'
- ☐ nicht definiert.

d) [1,5 Punkte] Welchen Wert hat `c` nach Ausführung der folgenden Anweisungen:

```
float a = 3.0, b = 1.7;
float c = a % b;
```

- ☐ 1.0
- ☐ 1.3
- ☐ 1.7
- ☐ 1.7647
- ☐ keinen, da die Operation nicht zulässig ist.

e) [1,5 Punkte] Welchen Dezimalwert hat `b` nach Ausführung der folgenden Anweisungen, wenn `char`-Werte 8 Bit lang sind:

```
typedef signed char s_char;
typedef unsigned char u_char;
s_char a = -72;
u_char b = (u_char)a;
```

- ☐ -72
- ☐ 72
- ☐ 183
- ☐ 184
- ☐ keinen, da die Operation nicht zulässig ist.

f) [1,5 Punkte] Welche Art von Fehler liegt im folgenden Codefragment vor?

```
int a = 7 >> 1;
int b = 7 / 2;
int c = 7 / (a-b);
```

- ☐ nur logischer Fehler
- ☐ nur syntaktischer Fehler
- ☐ nur semantischer Fehler
- ☐ syntaktischer und semantischer Fehler
- ☐ kein Fehler

Aufgabe 2: Zweierpotenzen *[9 Punkte]*

Schreiben Sie ein vollständiges C-Programm, das alle Zweierpotenzen $p = 2^i$ mit $1 \leq p \leq 1.000.000$ am Bildschirm ausgibt, und zwar in der Form

```
2 hoch 0 = 1
2 hoch 1 = 2
2 hoch 2 = 4
...
```

Dabei darf nur die Bibliothek `stdio.h` verwendet werden!

Aufgabe 3: Zeiger und Funktionen in C *[11 Punkte]*

a) [5 Punkte] Schreiben Sie eine Funktion `swap`, die den Inhalt zweier `int`-Variablen vertauscht.
Geben Sie ein Beispiel an, in dem diese Funktion aufgerufen wird.

b) [6 Punkte] Gegeben sei das folgende Programmfragment:

```
#define L1 5
#define L2 3

...

int main() {
  int A[L1][L2] = {{1,2,3},{2,3,4},{3,4,5},{4,5,6},{5,6,7}};
  int B[L2][L1];
  transpose(A,B);
  ...
}
```

Schreiben Sie eine geeignete Funktion `transpose`, die das Array `B` so vollschreibt, dass `B[i][j] = A[j][i]` gilt!
(Etwas mathematischer ausgedrückt, erstellt die Funktion `transpose` die transponierte Matrix `B` zur Matrix `A`.)

Aufgabe 4: Dynamische Datenstrukturen *[14 Punkte]*

Ein Binärbaum besteht aus Knoten, die höchstens zwei Nachfolger haben. In jedem solchen Knoten werde ein `unsigned int`-Wert gespeichert (siehe Abb. 1). Der Datentyp `knoten` (siehe Abb. 2) ist daher wie folgt definiert:

```
typedef struct node{
  unsigned int wert;          // Speichert den int-Wert
  struct node *kind1;         // Speichert den linken Nachfolger
  struct node *kind2;         // Speichert den rechten Nachfolger
} knoten;
```

Wenn ein Knoten keinen linken bzw. rechten Nachfolger hat, dann ist `kind1=NULL` bzw. `kind2=NULL`.

a) [6 Punkte] Schreiben Sie eine Funktion mit der Signatur

```
knoten* neuerKnoten(unsigned int, knoten*, knoten*),
```

die einen neuen Knoten anlegt und einen Zeiger auf diesen Knoten zurückliefert.
Tipp: Beachten Sie, dass dieser Datentyp dem Knoten in einer verketteten Liste sehr ähnlich ist.

b) [8 Punkte] Schreiben Sie eine <u>rekursive</u> Funktion mit der Signatur

```
unsigned int gibMaximum(knoten*),
```

die als Eingabe einen Zeiger auf die Wurzel eines Baumes erhält und die den größten Wert zurückgibt, der im Baum gespeichert ist. Dabei bleibt es Ihnen überlassen, in welcher Ordnung der Baum traversiert wird (inorder, preorder, postorder).
Tipp: Das Maximum in einem Binärbaum ist gleich dem Maximum aus dem Wurzelknoten und dem linken und rechten Teilbaum.

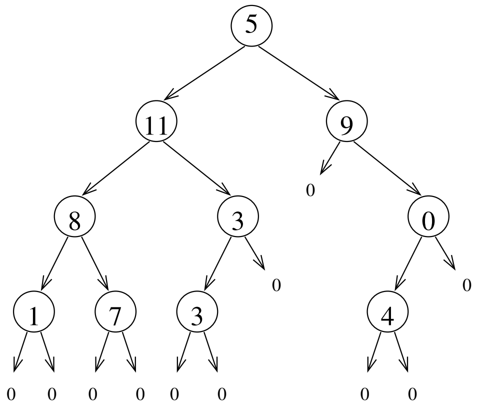

Abb. 1: Aufbau eines Binärbaums

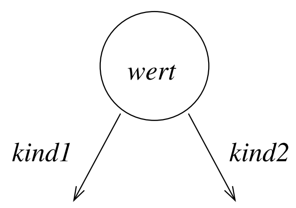

Abb. 2: Datenstruktur Knoten

(Platz zur Lösung von Aufgabe 4)

Aufgabe 5: Assembler-Programmierung (M 68000) *[11 Punkte]*

Schreiben Sie ein möglichst kurzes Programmstück in Assembler, das den Speicherbereich zwischen `$7000` und `$701F` wortweise durchsucht. Nach Abarbeitung des Programmstückes soll der größte Wortwert aus dem durchsuchten Bereich in Register `D0` und seine Adresse im Register `A0` stehen.
Bem.: Es ist <u>kein</u> vollständiges Assembler-Programm gefragt. Insbesondere müssen keine Assemblerdirektiven (wie `ORG $0` usw.) angegeben werden!

Aufgabe 6: Unterprogrammtechnik (M 68000) *[12 Punkte]*

a) [5 Punkte] Schreiben Sie ein Unterprogramm namens `SWAP-BYTE`, das die Werte der durch Register `A0` und `A1` adressierten Bytes im Hauptspeicher vertauscht.

b) [7 Punkte] Schreiben Sie ein Hauptprogramm, das die Funktion `SWAP` verwendet, um die Reihenfolge der Bytes im Speicherbereich zwischen `$7000` und `$701F` umzukehren. Wenn der Speicherbereich also vor Ausführung die Belegung

```
007000: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
007010: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
```

hatte, so soll er hinterher die Belegung

```
007000: AF AE AD AC AB AA A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
007010: 0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00
```

haben.