Physikalisch–Technische Lehranstalt Wedel
Prof. Dr. Uwe Schmidt

18. August 1997

Aufgaben zur Klausur **C** im SS 98 (IA 302)

Zeit: 60 Minuten erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 6 Seiten

---

Aufgabe 1:

Bei dem folgenden Programm wird angenommen, daß **long int** in 32 bit und **char** in 8 bit in der Maschine dargestellt werden.

```
#include <stdio.h>

int main(int argc, char * argv[]) {

    union {
        long int l;
        char c[4];
    } cv;

    cv.l = 0x01020304l;

    printf("%d,%d,%d,%d\n",
            cv.c[0],
            cv.c[1],
            cv.c[2],
            cv.c[3]);

    return 0;
}
```

Welches sind die möglichen Ausgaben für dieses Programm?

. . . . . .

. . . . . .

. . . . . .

Aufgabe 2:

Gegeben sei das folgende Programmstück.

```
char *p1, *p2;

void f(void) {

    *++p1 = *++p2;

    *p1-- = *p2--;
}
```

Schreiben Sie den Anweisungsteil der Funktion so um, daß keine Increment– und Decrement–Operatoren verwendet werden und nur eine Zuweisung pro Anweisung ausgeführt wird.

.....

.....

.....

.....

.....

.....

Aufgabe 3:

Gegeben seien die folgenden Variablen:

```
int i;
unsigned int u;
long int l;
double f;
char *cp;
int *ip;
int a[20];
```

Bestimmen Sie für die folgenden Ausdrücke den Typ. Vorsicht: Es kommen fehlerhafte Ausdrücke von. Kennzeichnen Sie diese entsprechend

i + f .....

++f .....

ip[i] .....

a[a[i]] .....

cp + 0x42 .....

cp ? i : f .....

f == 0 .....

! ip .....

~ip .....

cp && cp .....

f += i .....

~l .....

!l .....

l || i .....

l | i .....

Aufgabe 4:

Entwickeln Sie Makros zur Erzeugung von Bitmasken. Die Makros sollen Ausdrücke vom Typ **int** erzeugen. Sie sollen unabhängig von der Zahlendarstellung, 1–er oder 2–er-Komplement, sein.

1. Ein Makro `low_zeroes(n)` zum Setzen der `n` niederwertigen Bits auf 0, alle anderen Bits sollen auf 1 gesetzt werden.

   .....

   .....

   .....

2. Ein Makro
   low_ones(n) zum Setzen der `n` niederwertigen Bits auf 1, alle anderen Bits sollen auf 0 gesetzt werden.

   .....

   .....

   .....

3. Ein Makro
   mid_zeroes(width,offset), das die niederwertigen `offset` Bits auf 1 setzt, die folgenden `width` Bits auf 0, und alle übrigen wieder auf 1.

   .....

   .....

   .....

4. Ein Makro
   mid_ones(width,offset), das die niederwertigen `offset` Bits auf 0 setzt, die folgenden `width` Bits auf 1, und alle übrigen wieder auf 0.

   .....

   .....

   .....