Fachhochschule Wedel
Prof. Dr. Uwe Schmidt

1. Februar 2000

Aufgaben zur Klausur **C** im WS 99/00 (II 21)

Zeit: 90 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 7 Seiten

---

Aufgabe 1:

Gegeben seien die folgenden Variablen und Funktionen:

```
unsigned int x;
long int s;
float f;
unsigned char *p1;
int *p2;
void *p3;
int (*pf)(void);
void (*pf2)(double);
int f1(void);
int f2(int x1,int x2);
void f3(double x1);
```

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

*(p2+x) .....

p3+x .....

pf=f1() .....

*(p3+x) .....

p1 == p3 ? f1 : pf .....

~p2 .....

! p2 .....

*p1 && p1 .....

*p1 & p1 .....

s || x .....

pf2==f3 .....

s | x .....

(*pf)(f1()) .....

pf=f1 .....

pf2=f3 .....

f=f3(f) .....

*p3 .....

Aufgabe 2:

Gegeben sei das folgende Programm

```
#include <stdio.h>

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

    int i, sum=0;

    for(i=0; i<10; ++i) {
        switch (i) {
        case 0:
        case 3:
        case 4:
        case 7:
            sum += i+1;
        default:
            continue;
        case 6:
            break;
        }
        break;
    }

    printf("%d\n",sum);

    return 0;
}
```

Welche Ausgabe erzeugt dieses Programm?

.....

Aufgabe 3:

Entwickeln Sie Makros zur Erzeugung von Bitmasken. Die Makros sollen Ausdrücke vom Typ **unsigned 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.

   .....

   .....

   .....

Aufgabe 4:

Gegeben sei das folgende C–Programm zur Verarbeitung von Mengen als Bitstrings.

```
#include <stdio.h>

typedef unsigned char Menge;
#define MengeMax 8

void printMenge(Menge s) {
    unsigned int i = MengeMax;
    while ( i-- != 0 )
        printf("%1u", (unsigned int)((s >> i) & 1));
}

static unsigned int linecnt = 0;

#define PRINT(s) { printf("%2u) ", ++linecnt); printMenge(s); printf("\n"); }

#define einStueck(n,m) (dieErsten(m+1) ^ dieErsten(n))
#define dieErsten(n) (einElement(n) - 1)
#define einElement(i) ( (Menge)(1 << (i)) )

int main(void) {
    Menge s1;

    PRINT( einElement(0) );
    PRINT( einElement(MengeMax-1) );

    PRINT( (Menge)1 );
    PRINT( (Menge)17 );

    PRINT( einStueck(2,4) );
    PRINT( einStueck(7,1) );
    PRINT( einStueck(1,MengeMax) );

    PRINT( 21 & 20 );
    PRINT( 21 && 20 );

    s1 = ~einStueck(0,3) | einStueck(1,6); PRINT(s1);
    s1 = einStueck(1,6) ^ ((128 - 1) * 4); PRINT(s1);

    s1 = 12 + 48;
    s1 = s1 ^ (s1 & (~s1 + 1)); PRINT(s1);

    return 0;
}
```

Die Mengen sind in diesem Beispiel 8 Bits lang, können also die Elemente $0, 1, \ldots, 7$ enthalten. *printSet* gibt eine Menge im Binärformat aus. Die Menge, die nur die 1 enthält würde als 00000010 ausgegeben werden. Das *PRINT* Makro gibt jeweils eine Menge pro Zeile aus und numeriert die Zeilen durch.

Welche 12 Ausgabezeilen erzeugt dieses Programm?

1) .....

2) .....

3) .....

4) .....

5) .....

6) .....

7) .....

8) .....

9) .....

10) .....

11) .....

12) .....
