

---

Aufgaben zur Klausur C im WS 99/00 (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 5 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

$*(p2+x)$  .....

$p3+x$  .....

$pf=f1()$  .....

$*(p3+x)$  .....

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

$\sim p2$  .....

$! p2$  .....

$*p1 \&\& p1$  .....

$*p1 \& p1$  .....

$s || x$  .....

$pf2==f3$  .....

$s | x$  .....

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

$pf=f1$  .....

$pf2=f3$  .....

$f=f3(f)$  .....

$*p3$  .....

## 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.

.....  
.....  
.....