

---

Aufgaben zur Klausur **C** im WS 98/99 (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 sei das folgende Programm

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

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

    char *x[10];
    char **p = x;
    char **q = p;

    if ( strcmp(*p++, *q++) )
        printf("gleich");
    else
        printf("nicht gleich");

    return 0;
}
```

Welche Ausgabe erzeugt dieses Programm?

.....

Warum?

.....

.....

## **Aufgabe 2:**

Gegeben seien die folgenden Variablen:

**int** x;

**unsigned int** u;

**long int** s;

**float** f;

**char** \*p1;

**int** \*p2;

**void** \*p3;

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

- ++f .....
- p2[x] .....
- p1 ? x : f .....
- ! p3 .....
- p1 == p3 .....
- p1 = p3 .....
- ~p2 .....
- p1 && p1 .....
- x += f .....
- ~s .....
- ! s .....
- p3 ? x : f .....
- s || x .....
- p1 + 0x23 .....
- s | x .....

**Aufgabe 3:**

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.

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