
Aufgaben zur Klausur **C** im SS 97 (IA45)

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.

Bei allen C Routinen sollen die Parameter und Funktionsresultate mit ANSI-C Prototypen angegeben werden.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 5 Seiten

Aufgabe 1:

Gegeben sei das folgende C-Programm. Hinweis: der \ll Operator hat geringere Priorität als + und -.

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

#define twotimes(x) x + x
#define sqr(x) x * x
#define min(x,y) x < y ? x : y
#define squareroot(x) (++cnt, sqrt(x))

int cnt = 0;

int main(void)
{
    int i = 4, j = 3, r;
    double f = 1.25;

    r = twotimes(i << 2);
    printf(".1) r = %d\n",r);

    r = sqr(i + j);
    printf(".2) r = %d\n",r);

    r = 2 * twotimes(i);
    printf(".3) r = %d\n",r);

    r = sqr(twotimes(i));
    printf(".4) r = %d\n",r);

    r = twotimes(2*f);
    printf(".5) r = %d\n",r);

    r = min(twotimes(i+1),sqr(i));
    printf(".6) r = %d\n",r);

    r = min(squareroot(i+1),squareroot(j+1));
    printf(".7) r = %d, cnt = %d\n",r,cnt);

    return 0;
}
```

Welche 7 Ausgabezeilen erzeugt dieses Programm

.....

.....

.....

.....

.....

.....

.....

Ändern Sie das Makro *sqr* so ab, daß es für .2) den gewünschten Effekt hat.

.....

Ändern Sie das Makro *twotimes* so ab, daß es sich immer wie eine Funktion verhält

.....

Warum kann das Makro *min* nicht so verändert werden, das es sich immer, bei jedem Gebrauch, wie eine gleichwertige Funktion verhält?

.....

Warum kann der Gebrauch von Makros in der Form von *min* die Programmausführung verlangsamen (im Vergleich zu Funktionen).

.....

Aufgabe 2:

Die folgende C-Funktion testet, ob die Zeichenreihe $s1$ Anfangsstück der Zeichenreihe $s2$ ist.

```
int isPrefix(char * s1, char * s2)
{
    return
        *s1 == 0
        || ( *s1 == *s2
            &&
            isPrefix(s1+1, s2+1) );
}
```

Formen sie diese Funktion um in eine gleichwertige, die keine Funktionsaufrufe mehr enthält.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Aufgabe 3:

Welche der folgenden Ausdrücke sind equivalent zu dem Ausdruck $a[j][k]$

$$*(a[j] + k)$$

ja nein

$$** (a[j + k])$$

ja nein

$$*(a + j)[k]$$

ja nein

$$*(a + k)[j]$$

ja nein

$$*((*(a + j)) + k)$$

ja nein

$$** (a + j) + k$$

ja nein

$$*(&a[0][0] + j + k)$$

ja nein
