
Aufgaben zur Klausur C im WS 2002/03 (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:

Gegeben seien die folgenden Deklarationen und Initialisierungen

```
static int ar[] = {10, 15, 4, 25, 3, -4};  
int *p = &ar[2];
```

Welche Resultate werden durch die folgenden Ausdrücke berechnet?

1. $*(p + 1)$

.....

2. $p[-1]$

.....

3. $(ar - p)$

.....

4. $ar[*p - -]$

.....

5. $*(ar + ar[2])$

.....



Aufgabe 2:

Gegeben sei folgendes Pascal Programmstück

```
type
    Feld = array [1..10] of integer;

procedure tausch(var f1, f2 : Feld);
var
    tmp : Feld;
begin
    tmp := f1;
    f1 := f2;
    f2 := tmp
end ;
```

Übersetzen Sie die Typdefinition und die Prozedur in gleichwertige Datentypen und Routinen in C. Verwenden Sie keine Schleifen und kein *memcpy*. Beachten Sie, dass Zuweisungen an Variablen vom Typ *Feld* vorkommen.

(Ergebnis bitte auf der folgenden Seite)

Das gleichwertige C Programm

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



Aufgabe 3:

Gegeben ist das folgende C Programmstück. Dieses soll in eine header Datei *p.h* und eine Implementierungsdatei *p.c* aufgeteilt werden, so daß es von mehreren anderen Modulen verwendet werden kann. Kennzeichnen Sie durch Ankreuzen des *.h* Feldes für die header Datei oder *.c* für die Implementierungsdatei, in welche Datei die einzelnen Codestücke übertragen werden müssen.

| | | |
|---|-----------------------------|-----------------------------|
| <code>static int</code> nochEineZahl; | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>#include <stdio.h></code> | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>typedef unsigned long</code> Myint; | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>const</code> Myint maximum = 8888; | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>extern</code> Myint eineZahl; | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>typedef struct X * Px;</code> | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| Myint eineZahl = 42; | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>static int</code> nochEineZahl = 4711; | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>static int</code> f (Px x1); | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>int</code> g (Px x1); | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>struct X</code> { Myint d; } X; | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| Myint read (<code>struct X</code>); | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| Myint read (<code>struct X</code> t) { return t.d + nochEineZahl; } | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>#define</code> minimum 1111 | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>int</code> check (Px t) { return t->d \geq minimum; } | <input type="checkbox"/> .h | <input type="checkbox"/> .c |
| <code>void</code> outX (FILE * out, Px t); | <input type="checkbox"/> .h | <input type="checkbox"/> .c |

```
void outX (FILE * out, Px t) {  
    fprintf(out, "%1u\n", t→d);  
}
```

.h .c

```
extern Myint einPaarZahlen[];
```

.h .c

```
int g (Px x1) {  
    return f(x1) + 2;  
}
```

.h .c

```
static int f (Px x1) {  
    return read (*x1);  
}
```

.h .c

```
#define init_X(x) ((x).d = 0)
```

.h .c

```
struct s { int a[minimum]; };
```

.h .c

```
extern struct s einStruct;
```

.h .c