

Klausur zur Vordiplom-Prüfung Programmierungsmethodik

(Informatik: Ingenieurwesen)

13. Februar 2003

Sie haben 90 Minuten Zeit zum Bearbeiten der Klausur.

Tragen Sie bitte zunächst Ihren Namen, Ihren Vornamen, Ihre Matrikelnummer, Ihr Studienfach und Ihr Studiensemester in **DRUCKSCHRIFT** in die folgenden jeweils dafür vorgesehenen Felder ein.

Name:														
Vorname:														
Matr.-Nr.:								Fach					Sem.	

Aufg.	Punkte	Korr.
1		
2		
3		
4		
5		

Σ	
----------	--

Zur Beachtung:

Schreiben Sie bitte weder mit Bleistift noch mit Rotstift.

Kennzeichnen Sie bitte jedes Blatt mit Ihrem Namen und Ihrer Matrikelnummer.

Es sind keinerlei Hilfsmittel in dieser Klausur zugelassen!

Vergessen Sie nicht den „Vorbehalt“ zu unterschreiben.

Hinweis zum Programmieren:

Programmieren sie die Aufgaben in ANSI-C

Vorbehalt

Ich bin darüber belehrt worden, dass die von mir zu erbringende Prüfungsleistung nur dann bewertet wird, wenn die Nachprüfung durch das Zentrale Prüfungsamt der TUHH meine offizielle Zulassung vor Beginn der Prüfung ergibt.

(Datum, Unterschrift)

Aufgabe 1 (10 Punkte)

Schreiben Sie ein **lauffähiges** Programm, das einen Text aus einer Datei einliest und die eingegebenen Wörter zeilenweise in eine zweite Datei ausgibt. Zum Beispiel soll der Text

Nur du, Gudrun!

gespeichert werden als:

Nur

du,

Gudrun!

Das Programm soll von der Kommandozeile die Namen der Eingabedatei und Ausgabedatei übernehmen.

Hinweise: Das Programm braucht lediglich Leerzeichen in `\n` umzuwandeln und ansonsten die Eingabe Zeichen für Zeichen in die Datei ausgeben.

Aufgabe 2 (10 Punkte)

Ergänzen Sie zu dem folgenden Programm die Bildschirmausgabe:

```
#include <stdio.h>

int main()
{
    int x= 1, y= 2, z= 3, i, a[6], *p_1, *p_2;
    char ch;

    for(i= 0; i<6; i++) {
        a[i]= i;
        printf("a[%d]= %d (%p) ", i, a[i], a+i);
    }
    p_1 = &x;
    y = *p_1;
    *p_1= 3;
    printf("\n\n*** 1. Ausgabe ***\n");
    printf("x= %d y= %d ", x, y);
    printf("*p_1= %d\n", *p_1);

    x = 13;
    y = 5;
    ch = 'c';
    p_1= &z;
    z += ch - 'a' + *p_1;
    ch = ch + x/10;
    x %= 2 * (*(&y));
    p_2= a;

    p_1= p_2 + 3;
    *p_1= 99;
    *p_2= -11;
    printf("\n\n*** 2. Ausgabe ***\n");
    printf("x= %d z= %d ch= %c\n", x, z, ch);
    printf("p_1= %p p_2= %p\n", p_1, p_2);
    printf("a[0..5]= ");
    for(i= 0; i<6; i++) printf(" %3d", a[i]);

    for(i= 0; i<6; i++) a[i]= i;
    p_1= a;
    p_2= a + 5;
    for(i= 0; i<3; i++) {
        *p_1 = 2 - *p_2;
        *p_2+= i;
        p_1+= 2;
        p_2-= 2;
    }
    printf("\n\n*** 3. Ausgabe ***\n");
    printf("p_1= %p p_2= %p\n", p_1, p_2);
    printf(" a[0..5]= ");
    for(i= 0; i<6; i++) printf(" %3d", a[i]);

    return 0;
}
```

Die Bildschirmausgabe:

```
a[0]= 0 (0x22feb8)  a[1]= 1 (0x22feb8)  a[2]= 2 (0x22fec0)
a[3]= 3 (0x22fec4)  a[4]= 4 (0x22fec8)  a[5]= 5 (0x22fecc)
```

*** 1. Ausgabe ***

```
x= ---- y= ---- *p_1= ----
```

*** 2. Ausgabe ***

```
x= ---- z= ---- ch= ----
```

```
p_1= ----- p_2= -----
```

```
a[0..5]= ----
```

*** 3. Ausgabe ***

```
p_1= ----- p_2= -----
```

```
a[0..5]= ----
```

Aufgabe 3 (10 Punkte)

Zur Darstellung von Punkten und Kreisen in der Ebene seien die Strukturtypen wie folgt deklariert:

```
typedef struct {  
    double x;  
    double y;  
} PUNKT;
```

und

```
typedef struct {  
    PUNKT mittelpunkt;  
    double radius;  
} KREIS;
```

Schreiben Sie eine Funktion mit dem Prototypen

```
int enthalten ( PUNKT *p, KREIS *k);
```

die berechnet, in welcher Relation die per Zeiger übergebenen Daten stehen:

Liegt der referenzierte Punkt im Kreis (inklusive Rand), so soll der Funktionswert 1 sein, ansonsten 0.

Aufgabe 4 (10 Punkte)

Schreiben sie ein **lauffähiges** Programm, welches mit Hilfe von **Rekursion** die Quersumme (die Summe aller Ziffern) einer ganzen Zahl berechnet. Benutzen sie für ihren Algorithmus die Modulo-Teilung, um an die jeweils letzte Ziffer der Zahl zu gelangen. Diese wird dann zur Quersumme der vorderen Ziffer hinzuaddiert. Führen sie eine Fehlerbehandlung durch für den Fall eines negativen Eingabeargumentes.

Aufgabe 5 (10 Punkte)

Markieren und verbessern Sie die Fehler (**genau 10, syntaktisch und semantisch**) in dem untenstehenden Programm, welches von der Kommandozeile zwei ganze Zahlen und ein mathematisches Symbol einliest, die entsprechende Operation auf die beiden Ganzzahlen ausführt und das Ergebnis wieder ausgibt, d.h. etwa: `calc 2 + 3`. **Falsch erkannte Fehler werden negativ gewertet!**

```

1 #include <stdio.h>
2 #include <ctype.h>
3
4 int main(int argc, char argv[])
5 {
6     if(argc != 4) return;
7     a = str2int(argv[1]);
8     b = str2int(argv[3]);
9     switch( *argv[2] ){                                /* Fallunterscheidung für
10                                                         die Operatoren */
11         case '+' : printf("%d + %d = %d\n", a, b, a+b);
12                     break;
13         case '-' : printf("%d - %d = %d\n", a, b, a-b);
14                     break;
15         case '*' : printf("%d * %d = %d\n", a, b, a*b);
16                     break;
17         case '/' : printf("%d / %d = %d\n", a, b, a/b);
18                     break;
19         default : printf("unbekannter Operator %s\n", *argv[2]);
20     }
21     return 0;
22 }
23
24 int str2int(char *string)
25 {
26     int i, sign= 1, n= 0;
27     for(i= 0, isspace(string[i]); i++);
28     if( string[i] == '-') sign= -1;
29     if((string[i] == '+') || (string(i) == '-'))
30         i++;
31     for(; isdigit(string[i]); i++)
32         n = 10 * n + (string[i] - "0");
33     return sign*n;
34 }

```

[illegible]