

## Klausur zur Zwischen-Prüfung Programmiermethodik

(Informatik-Ingenieurwesen / Informationstechnologie)

8. Januar 2009

Sie haben 45 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.		

Es sind alle Aufgaben zu bearbeiten. Insgesamt können bis zu 28 Punkte erreicht werden.

Aufg.	Punkte	Korr.
1		
2		
3		

$\Sigma$	
----------	--

### Zur Beachtung:

Schreiben Sie bitte weder mit Bleistift noch mit Rotstift.

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 (8 Punkte)

Ein Student (Erstsemestler) sollte ein Programm schreiben, welches vom Benutzer zwei float Zahlen und einen Operator (+, -, \*, /) einliest, die entsprechende Operation ausführt und das Ergebnis auf dem Bildschirm ausgibt. Er sollte keine Fehlerbehandlung, außer Division durch Null, implementieren. Das Einlesen und die Berechnung sollten solange wiederholt werden, bis der Benutzer 'n' oder 'N' eingibt. Durch den schlechten Programmierstil des Studenten sind ihm einige Fehler unterlaufen. Den Meldungen des Compilers können wir entnehmen, dass genau 8 Fehler vorhanden sind.

Versuchen Sie alle Fehler zu finden.

**(Falsch erkannte Fehler werden negativ bewertet!)**

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      float a, b;
6      char again; op;
7
8      do {
9          printf("1. Operand: "); scanf("%f", &a);
10         printf("Operator: "); scanf("%c", &op);
11         printf("2. Operand: "); scanf("%f", &b);
12
13         switch (&op) {
14             case '+':
15                 printf("%f + %f = %f\n", a, b, a + b);
16                 break;
17
18             case '-':
19                 printf("%f - %f = %f\n", a, b, a - b);
20                 break;
21
22             case '*':
23                 printf("%f * %f = %f\n", a, b, a * b);
24
25             case '/':
26                 if (b == 0) {
27                     printf("Division durch 0.\n");
28                 }
29                 else {
30                     printf("%f / %f = %f\n", a, b, a / b);
31                 }
32                 break;
33             }
34
35         printf("Nochmal [j/n]? ");
36         scanf("%c", &again);
37     } while ((again != "n") && (again != 'N'));
38
39     return 0;
40 }
```

[illegible]

## Aufgabe 2 (8 Punkte)

Nehmen Sie bei dieser Aufgabe an, dass vorzeichenbehaftete Datentypen im 2-er Komplement dargestellt werden.

a) Stellen Sie die Zahlen 283, -475, 123, -3000, 728, -4356 als `short int` (binär) dar und addieren Sie diese.

283 :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
-475 :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
123 :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
-3000 :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
728 :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
-4356 :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
$\Sigma$	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

b) Geben Sie die kleinste und größte im 14-Bit-Zweierkomplement darstellbare Zahl in Dezimalschreibweise an.

kleinste Zahl: \_\_\_\_\_

größte Zahl: \_\_\_\_\_

c) Welche Ausgabe erzeugt folgende Code-Sequenz:

```
101    ...
102    short int k=-32768;
103    k=2*k;
104    printf("%d \n",k);
105    k=k-1;
106    printf("%d \n",k);
107    k=-32768;
108    k=k*(k+1)*(k+2);
109    printf("%d \n",k);
110    ...
```

---

---

---

Begründen Sie Ihre Antwort:

---

---

---

---

---

---

### Aufgabe 3 (12 Punkte)

Schreiben Sie ein Programm, das in einer Endlosschleife zwei ganze Zahlen  $n, k$  mit  $0 \leq k \leq n$  von der Tastatur einliest, den Binomialkoeffizienten  $\binom{n}{k}$  rekursiv in einer Funktion `binomial` berechnet und diesen abschließend ausgibt. Das Ergebnis von `binomial` soll vom Typ `int` sein. Dabei gilt folgende Rekursionsvorschrift:

$$\binom{n}{0} = 1, \quad \binom{n}{n} = 1, \quad \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad \text{für } 0 < k < n.$$

Um Zeit zu sparen, sollen für  $n \leq 100$  bereits berechnete Binomialkoeffizienten  $\binom{n}{k}$  in einem globalen Array `BiKo` abgelegt und sofern möglich in der Funktion `binomial` effizient verwendet werden. Das Feld `BiKo` soll im Hauptprogramm zu Beginn mit Nullen initialisiert werden. Eine Prüfung der Grenzen für `n` und `k` sei nicht erforderlich.

Schreiben Sie Ihr Programm hier rein !