Technische Universität Hamburg-Harburg
Arbeitsbereich Technische Informatik III
Prof. Dr. S.M. Rump und A. Hailer

WS 2003 / 04

Klausur zur Vordiplom-Prüfung Programmiermethodik

(Informatik Ingenieurwesen)

12. Februar 2004

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.			

Wenn Sie den Übungsschein für Programmiermethodik im WS 2003 / 04 erworben haben, können Sie in folgender Tabelle eine der Aufgaben ankreuzen, diese wird dann ohne Korrektur, d.h. ohne das diese von Ihnen bearbeitet wird, mit 10 Punkten bewertet.

Aufg.	Ersetzt durch Schein	Punkte	Korr.
1			
2			
3			
4			
5			
6			

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)

(a) Definieren Sie den Begriff "call by value":

(b) Welche Ausgabe hat folgendes C-Programm, begründen Sie Ihre Antwort!

```
#include <stdio.h>
int sum (int x, int y, int s) { s = x + y; return s; }
int diff (int *x, int *y, int *d) { *d = *x - *y; return *d; }

void main(void) {
  int x = 3, y = 2, s = 0, d = 0;
  printf("sum: %d   diff:  %d \n", sum(x, y, s), diff(&x, &y, &d));
  printf("s =  %d     d =  %d \n", s, d);
  }
```

(c) Ergänzen Sie folgendes Programmstück, bei dem die Funktion **swap** die Werte der Integer a und b vertauscht:

```
void swap (              a,              b)
  {









  }

int main (void) {
  int a = 3; int b = 4;
  /* Werte von a und b vertauschen */
  swap (              ,              );
  return 0;
  }
```

Aufgabe 2 (10 Punkte)

Ergänzen Sie die Ausgabe des folgenden Programms:

```
# include <stdio.h>
int main()
  {
  int x, y, z, i = 0, summe = 0;
  int *pa, *pb;
  printf( "&x= %x  &y=%x  &z = %x\n\n", &x, &y, &z);

  x = y = z = 3;   z = y * x++;
  printf( "x = %d  y = %d  z = %d\n\n", x, y, z);

  for(summe = i, pa = &i; i < 5; summe += (*pa)++);
  printf("summe = %d\n\n", summe);

  x = y = 1;
  pa = &x;  *pa= 2;
  pb = &z;  *pb= 4;
  z = x - y;
  pa = pb + 1;
  printf( "x = %d  y = %d  z = %d\n\n", x, y, z);

  printf( "pa = %x   pb = %x\n\n", pa, pb);
  printf( "*pa = %d  *pb = %d\n\n", *pa, *pb);

  return 0;
  }
```

Ausgabe des Programms:

```
&x= bffff6a8  &y=bffff6a4  &z = bffff6a0

x =        y =        z =

summe =

x =        y =        z =

pa =             pb =

*pa =            *pb =
```

Aufgabe 3 (10 Punkte)

Vervollständigen Sie das folgende C-Programm, das den Sinus einer Zahl $x \in [0, \frac{\pi}{4}]$ ausgibt. Wie unten bereits definiert soll als erster Test $x = \frac{\pi}{4}$ verwendet werden.

Der Sinus soll (i) mithilfe der entsprechenden Funktion aus der Bibliothek `math.h` und (ii) mit der Sinus-Reihe

$$\sum_{k=0}^{\infty}(-1)^k \frac{x^{2k+1}}{(2k+1)!}$$

berechnet werden. Hierbei soll die Summation erst dann abgebrochen werden, wenn sich die Summe nicht mehr ändert, die Anzahl der verwendeten Summanden ist auszugeben.

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

#define PI  3.1415926535897932


int main(void)
  {
  double x = PI/4;









  }
```

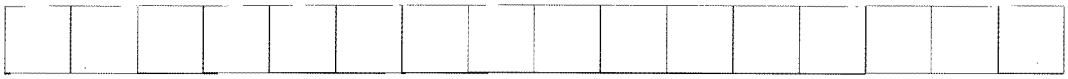
Die Ausgabe des Programms soll folgendes Format haben, wobei hier bereits die berechneten Resultate für $\sin(\frac{\pi}{4})$ und sowie die vom Programm ermittelte Anzahl der Summanden eingesetzt sind:

```
x = 0.785398
math.h:        sin(x) =  0.70710678118655
Taylor-Reihe: sin(x) =  0.70710678118655   (  9 Summanden)
```

Aufgabe 4 (10 Punkte)

(a) Stellen Sie die Zahlen 7 und -3 als `short int` dar:

7:



-3:

(b) Ergänzen Sie die Ausgabe des folgenden Programms:

```
#include<stdio.h>

int main()
  {
  short int a, b, w, x;
  long int c, d, y, z;


  a = 32767;  c = a;        /* 2^15 = 32768 */
  b = 65536;  d = b;        /* 2^16 = 65536 */

  w = a + 1; x = b + 1;
  y = c + 1; z = d + 1;

  printf("Short Int: a = %hd b = %hd: a+1 = %hd b+1 = %hd \n", a, b, w, x);
  printf("Long Int:  c = %ld d = %ld: c+1 = %ld d+1 = %ld \n", c, d, y, z);

  return 0;
  }
```

```
Short Int: a =          b =          : a+1 =          b+1 =

Long Int:  c =          d =          : c+1 =          d+1 =
```

(c) Beschreiben Sie kurz zwei Aufgaben des Präprozessors!

Aufgabe 5 (10 Punkte)

Beantworten Sie folgende Fragen in jeweils **ein oder zwei** Sätzen:

(a) Das folgende Programm und aufgezeichnete Ausführung zeigen merkwürdige Effekte:

```
#include<stdio.h>
int main (void)
  {
  int a[4];
  int j;

  for ( j=0; j<6; j++)
    {
    a[j] = 1;
    printf("j = %d     a[j] = %d \n", j, a[j]);
    }
  return 0;
  }
```

Ausgabe:

```
j = 0     a[j] = 1
j = 1     a[j] = 1
j = 2     a[j] = 1
j = 3     a[j] = 1
j = 1     a[j] = 1
j = 2     a[j] = 1
j = 3     a[j] = 1
j = 1     a[j] = 1
j = 2     a[j] = 1
j = 3     a[j] = 1
j = 1     a[j] = 1
j = 2     a[j] = 1
j = 3     a[j] = 1
j = 1     a[j] = 1
j = 2     a[j] = 1
  ⋮           ⋮
```

Was ist geschehen? Geben Sie eine kurze Erklärung dafür!

(b) Was ist eine rekursive Funktion?

(c) Welche Aufgabe erfüllt folgende Funktion?

```
int function (long bitset, int index)
  {
  return bitset & (1 << index)
  }
```

Aufgabe 6 (10 Punkte)

Ausgehend von folgender Datenstruktur ist eine lineare Liste aufgebaut, die alle Teilnehmer der Klausur mit der erreichten Gesamtpunktzahl als Elemente enthält:

```
struct element
  {
  char name[50];
  int punkte;
  struct element *next;
  };
```

Die nichtleere Liste ist nach Punkten (absteigend) sortiert und wird durch den Zeiger `*plist` repräsentiert. Schreiben Sie eine Funktion, die einen weiteren Studenten, der durch die Referenz `new` gegeben ist, an der richtigen Stelle in diese Liste einfügt! Fügen Sie dabei sinnvolle Kommentare ein.

```
void insert (struct element **plist, struct element *new)
  {




  }
```