Universität Erlangen-Nürnberg
Lehrstuhl für Betriebssysteme

Diplomvorprüfung Oktober 1999
"Grundlagen der Informatik"
5. Oktober 1999

Arbeitszeit:	120 Minuten
Erreichbare Punktzahl:	80

Angaben zur Person:

Name: ....

Vorname: ....

Geburtsdatum: .... Matrikelnummer: ....

Nicht von der Kandidatin bzw. vom Kandidaten auszufüllen!

Bewertung:

Aufgabe	1	2	3	4	5	6	7	8	Σ
max.	6	4	10	20	8	16	6	10	80
Punktzahl									

Note: ________________

Bemerkungen

- Jedes Klausurexemplar besteht aus 10 Seiten. Bitte prüfen Sie Ihr Exemplar auf Vollständigkeit und Druckqualität.
- Schreiben Sie die Lösungen zu den Aufgaben in die dafür freigehaltenen Zwischenräume auf den entsprechenden Seiten bzw. auf die unbedruckten Blattrückseiten. Sollte dieser Platz nicht ausreichen, können Sie sich von der Aufsicht zusätzliche **Leerblätter** in Ihr Klausurexemplar einheften lassen, **bevor Sie diese beschreiben**. Beschriebene Blätter und Schmierpapier können nicht eingeheftet werden und dürfen am Ende der Bearbeitungszeit nicht mit abgegeben werden.
- Versehen Sie alle zusätzlich eingehefteten Blätter mit Ihrem Namen und Ihrer Matrikelnummer.
- Schmierpapier liegt an Ihrem Platz bereit. Falls Sie zusätzliches Schmierpapier benötigen, bitte melden!
- Bevor Sie mit der Bearbeitung der Aufgaben beginnen:
  - Füllen Sie bitte die “Angaben zur Person” auf dem Deckblatt *gut leserlich* aus.
  - Legen Sie bitte Ihren Lichtbildausweis und Studentenausweis für die Kontrolle bereit.
  - Unterschreiben Sie bitte die folgende Erklärung.
- Als Hilfsmittel dürfen (außer Schreibmaterial) ein nicht programmierbarer Taschenrechner, sowie das Hilfsblatt “Assemblercodes” verwendet werden.
- Fragen zu den Prüfungsaufgaben können grundsätzlich **nicht** beantwortet werden.
- Sie dürfen den Raum nicht verlassen, **bevor** Ihre Personalien überprüft wurden.
- Bleiben Sie bitte am Ende der Bearbeitungszeit solange an Ihrem Platz sitzen, bis alle Klausurunterlagen eingesammelt sind.
- Falls Sie die Prüfung nach Erhalt der Prüfungsaufgaben aus gesundheitlichen Gründen abbrechen müssen, so muß Ihre Prüfungsunfähigkeit durch eine Untersuchung in der Universitätsklinik nachgewiesen werden.

Viel Erfolg!

Erklärung

Durch meine Unterschrift bestätige ich

- den Empfang der vollständigen Klausurunterlagen (10 Blätter)
- die Kenntnisnahme der obigen Informationen.

Ich bin damit einverstanden, daß mein Prüfungsergebnis unter Angabe der Matrikelnummer veröffentlicht wird: **JA** ☐ **NEIN** ☐

Erlangen, 5. Oktober 1999 ....
(Unterschrift)

Aufgabe 1 (6 Punkte)

a) Konvertieren Sie die Zahl

$$56026_{10}$$

zunächst durch Anwendung der *Divisionsmethode* in das Dualsystem und von dort durch *Blockbildung* in das Oktal- und Sedezimalsystem.

b) Konvertieren Sie die Zahl

$$0.4125_{10}$$

durch Anwendung des *Quellverfahrens für rationale Zahlen* in das Dual- und Oktalsystem. Konvertieren Sie anschließend aus der Dualdarstellung durch *Blockbildung* in das Sedezimalsystem.

Aufgabe 2 (4 Punkte)

Berechnen Sie im Dualsystem (der Rechenweg muß dargestellt werden!):

$$(11011001_2 + 11001_2) * 11101_2$$

Aufgabe 3 (10 Punkte)

Geben Sie die Inhalte der Speicherzelle `res`, der benötigten Register und des Kellerspeichers während und nach Ausführung des folgenden Assemblerprogramms an (in Dezimaldarstellung)!

```
 0          LAD,    @inp1
 1          SPI,    @par1
 2          LAD,    @inp2
 3          SPI,    @par2
 4          JSR,    @rmdr
 5  par1:   BLOCK,  1
 6  par2:   BLOCK,  1
 7   res:   BLOCK,  1
 8          LAD,    @res
 9          HLT
10  rmdr:   POPAC
11          LDI
12          ADD,    #3
13          PUSHAC
14          LAD,    %0
15  loop:   JEZ,    @bye
16          SUB,    %1
17          JGZ,    @loop
18          ADD,    %1
19   bye:   SPI,    %2
20          RET
21  inp1:   DATA,   #31
22  inp2:   DATA,   #6
```

Zeile	res	Akkumulator	Indexregister	Kellerspeicher

Aufgabe 4 (20 Punkte)

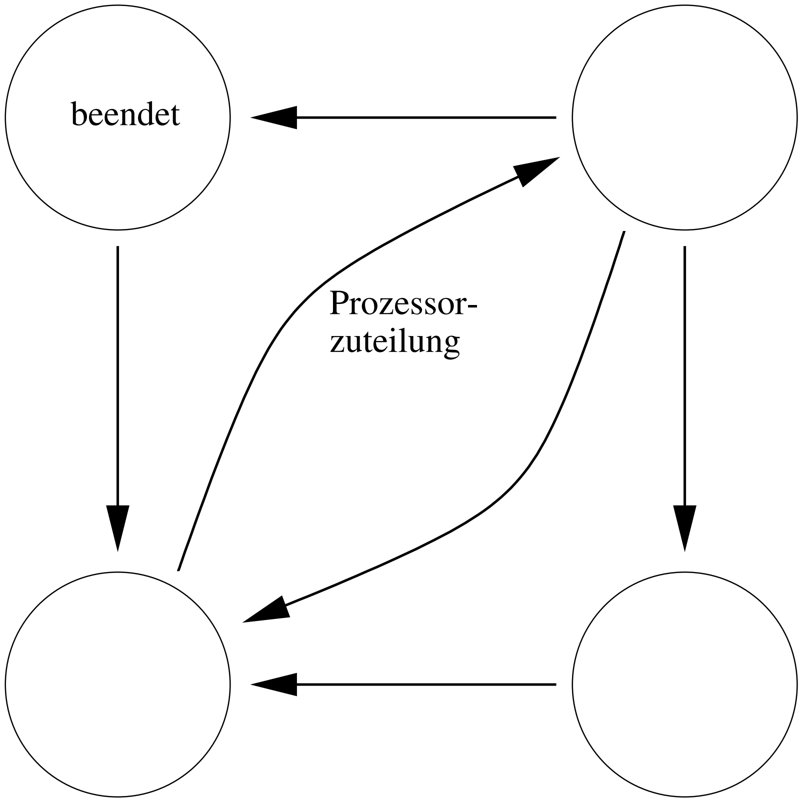
Übersetzen Sie das folgende `C`-Programm zur rekursiven Berechnung von Fakultäten ganzer Zahlen in die von Rembold angegebene hypothetische Assemblersprache! Verwenden Sie dabei für die Übergabe des Parameters an das Unterprogramm '`fak`' sowie für die Übergabe des Rückgabewertes an das Hauptprogramm den Kellerspeicher (Stack)! Desweiteren existiere ein Pseudo-Assembler-Befehl '`_input, @n`', welcher einen vom Benutzer über die Tastatur einzugebenden Wert in die Speicherzelle '`n`' lädt, sowie ein weiterer Pseudo-Assembler-Befehl '`_print, @n`' der den Inhalt der Speicherzelle '`n`' auf dem Bildschirm ausgibt.

```
# include <stdio.h>
int fak(int x)
{
    if(x==0)
       return 1;
    if(x==1)
       return 1;
    return x * fak(x-1);
}

void main(void)
{
   int n, result;
   do {
      scanf("%d", &n);
      if (n < 0) break;
      result = fak(n);
      printf("%d\n", result);
   } while(1);
}
```

Aufgabe 5 (8 Punkte)

Tragen Sie in das unten abgebildete, unvollständige Prozeßzustandsdiagramm die drei fehlenden Prozeßzustände sowie die fünf fehlenden Zustandsübergänge ein:



Aufgabe 6 (16 Punkte)

Geben Sie für das folgende Programm die *Gültigkeits-* und *Zugriffsbereiche* aller verwendeten Variablen an. Unterscheiden Sie gleich benannte Variablen durch Indizierung mit der Nummer des Blockes, in dem sie definiert werden.

```
BEGIN                                   (* Block 1 *)
   VAR a, c, d: INTEGER;
   PROCEDURE g( b, c: INTEGER )    (* Block 2 *)
      VAR a, d: INTEGER;
      d := 2*c+b;
      WHILE (d >= 0) DO
         BEGIN                          (* Block 3 *)
            VAR e: INTEGER;
            e := 3*c;
            b := c/2;
            d := d-1
         END;
      END;
      d := 3*b
   END;
   PROCEDURE h( a, b: INTEGER )    (* Block 4 *)
      VAR c, d: INTEGER;
      d := 4; b := 3; a := d+b;
      WHILE (a >= 0) DO
         BEGIN                          (* Block 5 *)
            VAR b: INTEGER
            b := d*a;
            a := a-1
         END;
      END;
   END;
   IF (a-d < 0) THEN
      BEGIN                             (* Block 6 *)
         VAR a: INTEGER;
         a := d/2;
         h( a, d )
      END;
   ELSE
   BEGIN
      VAR b: INTEGER;
      b := a-2;
      g( b, d )
   END;
END.
```

Aufgabe 7 (6 Punkte)

Entscheidungstabellen dienen zur Darstellung von Beziehungen zwischen Anforderungen und durchzuführenden Aktionen.

Gegeben sei die informelle Beschreibung einer Selbstbedienungs-Etikettierwaage aus der Obst- und Gemüseabteilung eines Supermarktes:

> Ist der Vorrat an selbstklebenden Etiketten in der Waage aufgebraucht, so wird der Kunde durch den auf der Anzeige erscheinenden Text „Bitte Personal verständigen!“ auf den nicht betriebsbereiten Zustand der Waage hingewiesen. Sind noch Etiketten vorhanden, legt der Kunde die zu wiegende Ware auf die Waagschale. Unterschreitet das Gewicht der aufgelegten Ware das Mindestgewicht der Waage von 20g, so wird dies dem Kunden auf der Anzeige der Waage mitgeteilt („Mindestgewicht unterschritten!“). Ebenso zeigt die Waage an, wenn das Warengewicht den Meßbereich der Waage von 5kg überschreitet („Höchstgewicht überschritten!“). Ist das Warengewicht innerhalb des zulässigen Meßbereichs, drückt der Kunde anschließend die zu der aufgelegten Ware passende Taste auf dem Bedienfeld der Waage. Da die Waage über 80 Produkttasten verfügt, im Supermarkt aber nur 60 verschiedene Obst- und Gemüsesorten angeboten werden, sind folglich 20 Tasten unbelegt. Wählt der Kunde fälschlicherweise eine der nicht belegten Tasten, so wird er durch einen entsprechenden Text in der Anzeige auf die Fehlbedienung hingewiesen („Wahl ungültig!“). Hat der Kunde eine gültige Taste gewählt, wird das gewünschte Etikett ausgedruckt.

Gesucht ist eine Entscheidungstabelle für die beschriebene Waage.

Aufgabe 8 (10 Punkte)

Im Arbeitsspeicher eines Rechensystems stehen zum Zeitpunkt $t = 0$ die Prozesse P_1, P_2 und P_3 in der unten gezeigten Anordnung.

Betriebs-system	P_1 $15M$	frei $15M$	P_2 $15M$	frei $12M$	P_3 $9M$	frei $18M$
12	27	42	57	69	78	96

Zu den folgenden Zeitpunkten ergeben sich Änderungen:

Zeitpunkt	Aktion	Speicherbedarf
$t = 1$	neuer Prozeß: P_4	$12M$
$t = 2$	neuer Prozeß: P_5	$18M$
$t = 3$	neuer Prozeß: P_6	$15M$

Lassen sich diese Speicheranforderungen erfüllen, wenn

a) die "Worst-Fit" Methode verwendet wird?

b) die "Best-Fit" Methode verwendet wird oder

c) die "First-Fit" Methode verwendet wird oder

Zeichnen Sie in jedem Fall die resultierende Speicherbelegung (entweder am Schluß oder zu dem Zeitpunkt, an dem die erste Speicheranforderung nicht mehr befriedigt werden kann)! Verwenden Sie dazu die vorbereiteten Belegungspläne:

a)

Betriebs-system	P_1 $15M$		P_2 $15M$		P_3 $9M$	
12	27	42	57	69	78	96

b)

Betriebs-system	P_1 $15M$		P_2 $15M$		P_3 $9M$	
12	27	42	57	69	78	96

c)

Betriebs-system	P_1 $15M$		P_2 $15M$		P_3 $9M$	
12	27	42	57	69	78	96

Welche Maßnahme kann das Betriebssystem durchführen, falls eine Speicheranforderung im obigen Beispiel nicht erfüllt werden kann?