Universität Erlangen-Nürnberg
Lehrstuhl für Informatik 4 (Verteilte Systeme und Betriebssysteme)

Diplomvorprüfung September 2001
"Grundlagen der Informatik"
27. September 2001

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	6	10	20	10	8	10	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, 27. September 2001 ....
(Unterschrift)

Aufgabe 1 (6 Punkte)

a) Konvertieren Sie die Zahl

$$38905_{10}$$

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

b) Konvertieren Sie die Zahl

$$0.7777_{10}$$

durch Anwendung des *Quellverfahrens für rationale Zahlen* in das Oktalsystem. Brechen Sie die Berechnung nach dem Ermitteln der neunten Nachkommastelle der oktalen Darstellung ab. Konvertieren Sie die Näherung anschließend unmittelbar in das Dualsystem und danach durch *Blockbildung* in das Sedezimalsystem.

Aufgabe 2 (6 Punkte)

a) Berechnen Sie die Differenz

$$73_8 - 37_8$$

durch Konvertieren der Zahlen ins Dualsystem und Ermittlung der Differenz im Dualsystem durch Komplementbildung und Konvertierung des Ergebnisses ins Oktalsystem (der Rechenweg muß dargestellt werden!)
b) Berechnen Sie

$$(28_{16} + 16_{16}) \cdot 7_{16}$$

durch Konvertieren der Zahlen ins Dualsystem und Ermittlung des Ergebnisses im Dualsystem und anschliessende Rückkonvertierung ins Sedezimalsystem.

Aufgabe 3 (10 Punkte)

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

```
0          LAD,    @num1
1          SPI,    @off0
2          JSR,    @subr1
3   off0:  BLOCK,  1
4   resu:  BLOCK,  1
5   num1:  DATA,   #6
6   num2:  DATA,   #33
7          LAD,    @resu
8          HLT
9   subr1: POPAC
10         LDI
11         ADD,    #4
12         PUSHAC
13         JSR,    @subr2
14         RET
15  subr2: LAD,    @num2
16  loop:  JEZ,    @ade
17         SUB,    %0
18         JGZ     @loop
19         ADD,    %0
20         INK
21   ade:  SPI,    %0
22         RET
```

Zeile	resu	Akkumulator	Indexregister	Kellerspeicher	off0

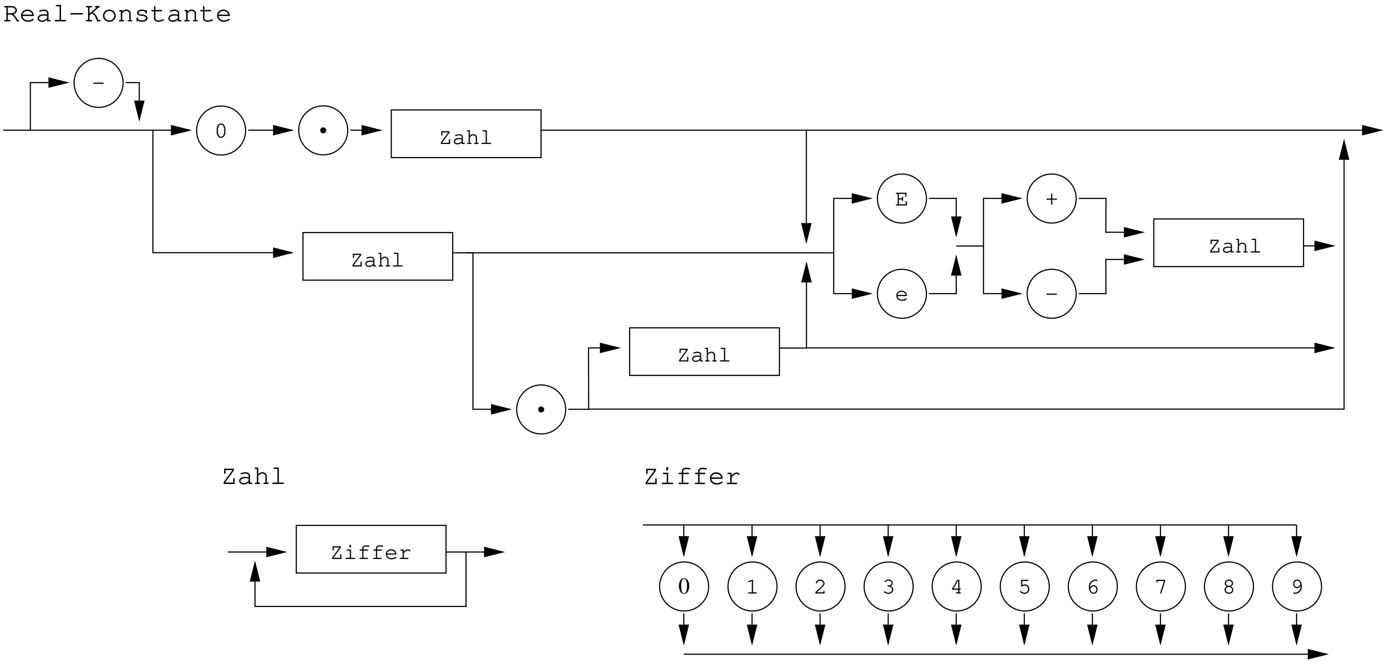
Aufgabe 4 (20 Punkte)

Übersetzen Sie das folgende `C`-Programm in die von Rembold angegebene hypothetische Assemblersprache! Verwenden Sie dabei für die Übergabe der Parameter an die Unterprogramme '`lcm()`' und '`gcd()`' den Kellerspeicher (Stack). Die Rückgabewerte der Unterprogramme sollen dem aufrufenden Programmteil im Akkumulator übergeben werden. Gehen Sie davon aus, dass in der Assemblersprache zur Ein-/Ausgabe der Werte die Pseudobefehle '`_input`' und '`_print`' zur Verfügung stehen.

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

int gcd(int a,int b)
{
   if (b==0) return a;
    return gcd(b,a % b);
}

int lcm(int x,int y)
{
   int g, aog;
   g=gcd(x,y);
   aog = x/g;
   return y*aog;
}

int main()
{
   int i,j, erg;
   printf("i = ");
   scanf("%d",&i);
   printf("j = ");
   scanf("%d",&j);
   if (j > i) erg = lcm(j, i);
   else erg = lcm(i, j);
   printf("lcm(%d, %d) = %d\n", i,j, erg);
   return 0;
}
```

Aufgabe 5 (10 Punkte)

Gegeben seien die folgenden Syntaxdiagramme :



a) Welche der unten aufgeführten Zeichenfolgen sind demnach gültige Real-Konstanten? (Unterstreichen!) Falls die Zeichenfolge keine gültige Real-Konstante ist, markieren S ie das längste Teilstück von Beginn an, das noch ableitbar ist.

i) `.692E-73` ii) `397.34e4+20` iii) `-3.14115` iv) `59.22E-12`

b) Geben Sie eine kontextfreie Grammatik $G = (N, T, P, S)$ an, die als Sprache L_G die Menge aller korrekten Klammerausdrücke erzeugt, d.h. $T = \{(,)\}$, jede linke Klammer hat eine entsprechende rechte Klammer und Paare zusammengehörender Klammern sind richtig eingebettet. So ist z.B. $(()())() \in L_G$ nicht aber $(()()))(()$.

Aufgabe 6 (8 Punkte)

Für die Fibonacci Zahlen gilt:

$$fib(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ fib(n-1) + fib(n-2) & n > 1 \end{cases}$$

Geben Sie jeweils ein `C`-Programm an zur Berechnung der Fibonacci-Zahlen

a) Durch Iteration
b) Durch Rekursion

an. Aus welchem Grund ist in diesem Fall die iterative Lösung der Rekursiven vorzuziehen?

Aufgabe 7 (10 Punkte)

Was versteht man unter den Begriffen *Objekt*, *Klasse*, *Attribut* und *Verhalten* im Rahmen der objektorientierten Programmierung. Formulieren Sie Ihre Antwort in kurzen Sätzen bzw. in Stichworten!

Aufgabe 8 (10 Punkte)

a) Beschreiben Sie das Konzept „Semaphor“ zur Prozeßsynchronisation, insbesondere die Funktion der beiden auf Semaphor-Variable und Prozesse wirkenden Operationen `P( ... )` und `V( ... )`.

b) Formulieren Sie den exklusiven Zugriff zweier Prozesse `A` und `B` auf einen „kritischen Abschnitt“ (wechselseitiger Ausschluss) mithilfe des Semaphors `mutex`.