7. Übungsblatt:
Programmierpraktikum II (SS 2002)

Abgabe: 15. Juli 2002

Aufgabe 1: Assembler-Direktiven

3 Punkte

Gegeben sei das folgende Assembler-Programm, das eine bestimmte mathematische Funktion berechnet:

```
        ORG     $0
        DC.L    $8000
        DC.L    START

        ORG     $2000
START   MOVE.W  BASE,D0
        MOVE.W  INPUT,D1
        MOVE.W  BASE,D2
        MOVE.W  #0,D3
LOOP    CMP.W   D2,D1
        BLT     ENDE
        DC.W    $5243
        MULU    D0,D2
        BRA     LOOP
ENDE    MOVE.W  D3,OUTPUT
        BREAK

        ORG     $7000
INPUT   DC.W    4321            ; Immer eine positive Zahl
BASE    DC.W    2               ; Immer eine positive Zahl
OUTPUT  DC.W    0
        END
```

Finde heraus, was das Programm leistet (natürlich kannst du dazu auch die beiden Input-Variablen INPUT und BASE verändern). Beschreibe die Funktionsweise (z.B. durch Angabe der berechneten Funktion) und schreibe Kommentare an die einzelnen Befehlszeilen. Erläutere dabei insbesondere, welche Aufgabe die einzelnen Assembler-*Direktiven* haben.

Aufgabe 2: Bedingte Verzweigungen

5 Punkte

Gegeben seien die folgenden Register- bzw. Speicherbelegungen:

```
   (D0) = $00014555
   (D1) = $000242FF
($1000) = $44
($1001) = $44
($1002) = $44
($1003) = $44
```

Welche der folgenden bedingten Verzweigungen würden jeweils ausgeführt? Erkläre jeweils das Verhalten des Branch-Befehls. Behandle jede Teilaufgabe als eigenständiges Problem!

a)
```
CMP.L  D0,D1
BGE    LABEL
```

b)
```
CMP.W  D0,D1
BLT    LABEL
```

c)
```
MOVE.B D1,$1003
BPL    LABEL
```

d)
```
CMP.L  $1000,D0
BGT    LABEL
```

e)
```
ADD.W  D0,D1
BVC    LABEL
```

Aufgabe 3: Unterprogramme

7 Punkte

Schreibe ein Programm, das die Bytes im Speicherbereich $7000 bis $701F aufsteigend sortiert.

- Schreibe zunächst ein Unterprogramm `sort`, das als Input zwei Adressen `A` und `B` im Speicher erhält und deren Inhalte vertauscht, falls `(A)` > `B` ist.
- Benutze den Algorithmus Bubble-Sort, um den Speicherbereich zu sortieren. Dieser soll das Unterprogramm `sort` benutzen und wie folgt funktionieren:

```
for i=$701E downto $7000
  for j=$7000 to i
    sort(Mem[j],Mem[j+1])
```

Bsp.: Wenn der Speicherbereich vor Ausführung des Programmes die Belegung

```
007000: 1a 4f 33 10 0c a2 35 67 01 18 bb 95 78 9d 10 18
007010: 5a 20 12 3e 9e 72 08 94 c1 d0 86 7b 97 00 e3 53
```

hatte, dann muss er hinterher wie folgt aussehen:

```
007000: 86 94 95 97 9d 9e a2 bb c1 d0 e3 00 01 08 0c 10
007010: 10 12 18 18 1a 20 33 35 3e 4f 53 5a 67 72 78 7b
```