10. Übungsblatt Informatik II
(Abgabe: 29.06.2005)

1. **Aufgabe:** ()

2. **Aufgabe:** (Ausdrucksauswertung)

   a.) Es reicht nicht das die Routine nur mit positiven Zahlen arbeitet. Das Multiplizieren funktioniert zwar für positive wie für negative Zahlen gleich, allerdings wird beim Algorithmus das b $(a-1)$-mal auf b aufaddiert. Wenn $a = 0$ terminiert der algorithmus. Ist a negativ dann würde der Algorithmus nicht terminieren, falls das nicht überprüft wird.

   b.) Der Algorithmus funktioniert genau wie der Algorithmus von Blatt5 Aufgabe 4. Es müssen nur die Register angepasst werden.

   c.)

   0:add(j, R0, k) ; Lade Basisaddresse von y in Reg j

   4:add(R1, R0, l) ; Lade "3ïn Register R1

   8:beqz(R1, 16) ; falls R1 enthält null, spring zum Ende

   12:addi(j, R0, 4) ; Addiere 4 zum Wort in Register j

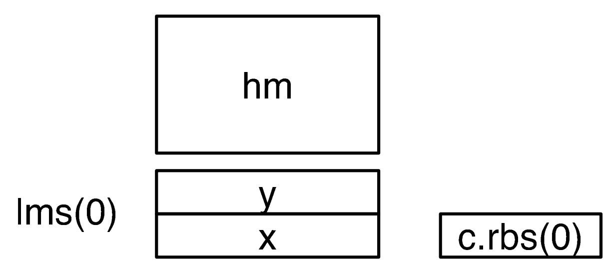
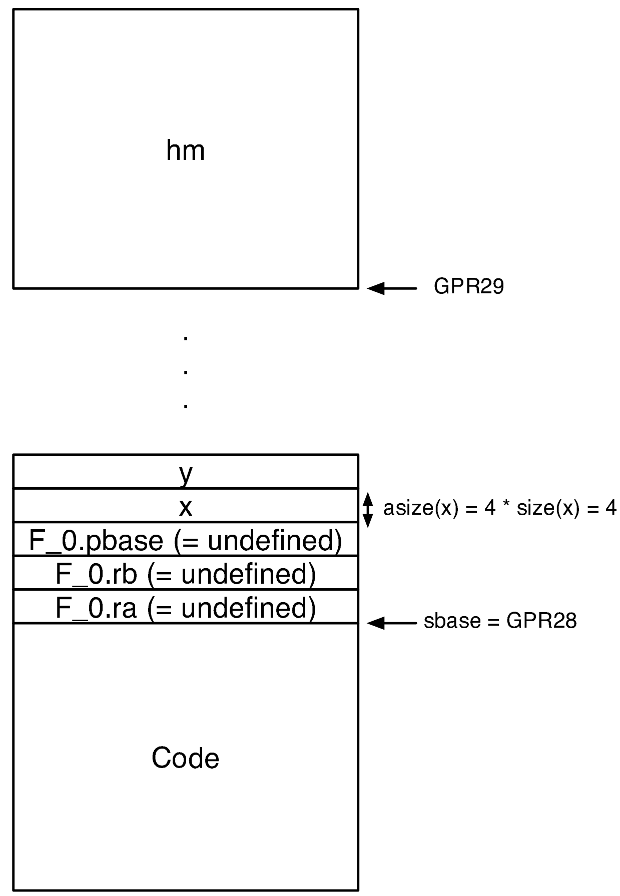
   16:subi(R1, R1, 1) ; dekrementiere Wert in R1

   20:j(-12) ; zum Schleifenanfang

   24:lw(j, j, 0) ; j erhält neuen Wert von x

3. **Aufgabe:**

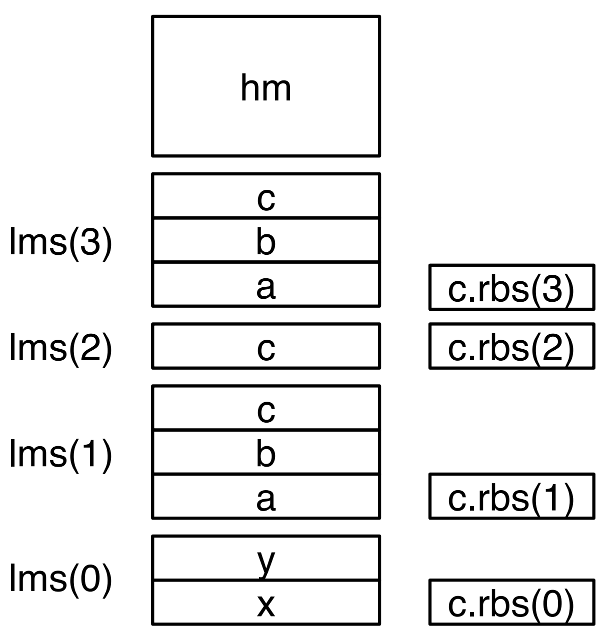
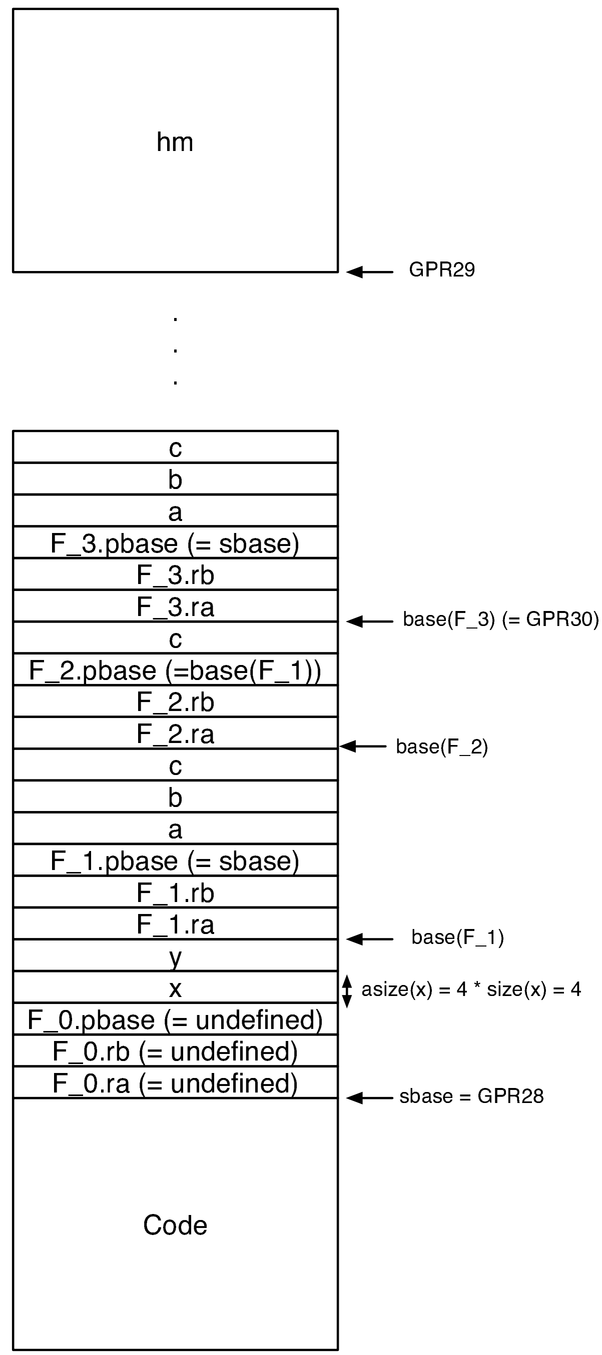
   a)

c^0:	d(0):
	

hm
lms(0)
y
x
c.rbs(0)

hm
GPR29
y
x
asize(x) = 4 * size(x) = 4
F_0.pbase (= undefined)
F_0.rb (= undefined)
F_0.ra (= undefined)
sbase = GPR28
Code

b)

c^0:	d(0):
	

4. **Aufgabe:**

Ansatz wie im Skript auf Seite 37:
c.pr: while e do {a} mit e = $n \neq 0$ und a = $result = result + n; n = n - 1$
Annahme: anfangs $n \neq 0$, da sonst kein Schleifendurchlauf.

$\Rightarrow va(c, e) = true$
$\Rightarrow c'.pr: \underbrace{a}_{1.Durchlauf}; \underbrace{while\ e\ do\ \{a\}}_{N-1 Durchlaeufe}$
Induktionsbehauptung: Sei $va(c^{\alpha}, result) = 0$ und $va(c^{\alpha}, n) = N$
$c^{\alpha}.pr: \ while\ e\ do\ a$
Da a aus 2 Anweisungen besteht und ein Schritt zum Auswerten von e benötigt wird, gilt nach erstem Durchlauf: $c^{\alpha+3}$
$\Rightarrow$ nach N Durchläufen: $c^{\alpha+3*N}$

$\Rightarrow va(c^{\alpha+3*N}, result) = 0 + N + N - 1 + \cdots + 1$
$va(c^{\alpha+3*N}, n) = 0$
$c^{\alpha+3*N}.pr: \ while\ e\ do\ a$

Im nächsten Schritt würde e zu false ausgewertet werden und der Programmrest ist ϵ
Nun kann man einen Induktionsbeweis über N (= Anzahl der Durchläufe) führen:

I.A.:
N=0:$va(c^{\alpha+3*N}, result) = va(c^{\alpha}, result) = 0$
$va(c^{\alpha+3*N}, n) = va(c^{\alpha}, n) = 0$
$c^{\alpha+1}.pr: \epsilon$
Die Gauß'sche Summe ist trivialer Weise für N=0 0
N=1:$va(c^{\alpha+3*N}, result) = va(c^{\alpha+3}, result) = 0 + 1 = 1$
$va(c^{\alpha+3*N}, n) = va(c^{\alpha+3}, n) = 1 - 1 = 0$
$c^{\alpha+3+1}.pr: \epsilon$
Die Gauß'sche Summe ist auch für N=1trivial gleich 1
I.B.: Für die letzten N-1 Durchläufe gelte:

$$va(c^{\alpha+3*(N-1)}, result) = \sum_{i=0}^{N-1} i$$
$$va(c^{\alpha+3*(N-1)}, n-1) = 0$$

I.S.: $N - 1 \rightarrow N$

$$va(c^{\alpha+3*N}, result) = N + va(c^{\alpha+3*(N-1)}, result) \overset{I.V.}{=} N + \sum_{i=0}^{N-1} i = \sum_{i=0}^{N} i$$
$$va(c^{\alpha+3*N}, n) = va(c^{\alpha+3*(N-1)}, n) - 1 = va(c^{\alpha+3*(N-1)}, 1) + va(c^{\alpha+3*(N-1)}, n-1) - 1$$
$$\overset{I.V.}{=} 1 + 0 - 1 = 0$$

$c^{\alpha+3*N+1}.pr := \epsilon$, da $n = 0$