

## Lista de Exercícios — Software Básico

1. Simule a execução de cada um dos trechos de código abaixo. Para cada instrução executada, indique claramente (1) qual instrução foi executada, (2) quais registradores tiveram seus valores alterados (considere apenas os registradores convencionais,  $\$R0$ – $\$R31$ ), e (3) quais os novos valores (*em hexadecimal*) contidos nos registradores alterados.

```
(a) init:  li      $a0, -8
          addi   $a1, $a0, 12
          bgtu  $a0, $a1, L1
          li    $a0, 0
L1:      move  $a1, $a0
          addi  $a1, $a1, -9
```

```
(b) init:  lui    $t0, 10
          addi  $t0, $t0, 16
          sub   $t0, $r0, $t0
```

- (c) Para o trecho abaixo, assuma que o label `str1` vale `0x100400A0`, e lembre-se que o valor ASCII do caracter `a` é 97 (decimal).

```
        .data
str1:   .asciiz "abafa"
        .text
init:   la     $s0, str1
        addi  $s1, $s0, 3
        lb   $t0, ($s1)
```

2. Implemente cada uma das funções abaixo em assembler MIPS. Todas as funções recebem o primeiro argumento no registrador  $\$a0$ ; se houver um segundo argumento, ele estará no registrador  $\$a1$ . Eventuais valores retornados devem ser colocados em  $\$v0$ . Além disso, as funções só podem “sujar” os valores dos registradores  $\$t1$ – $\$t9$ .

- (a) Uma função que recebe o endereço de um array de inteiros, e seu número de elementos, e retorna a soma dos elementos do array.
- (b) Uma função que recebe o endereço de um array de inteiros, e seu número de elementos, e retorna quantos elementos do array têm valor maior que zero.
- (c) Uma função que recebe o endereço de um array de inteiros, e seu número de elementos, e preenche cada elemento do array com duas vezes seu índice (isso é,  $a[i] = i*2$ ;