

Exercícios 01

1. Utilizando os mnemônicos LOAD (carga de dados), STORE (armazenamento de dados), ADD (soma de dados) e SUB (subtração de dados) crie um segmento de código, em linguagem assembly, que realize as operações abaixo. As instruções de movimentação de dados utilizam 2 operandos e as instruções aritméticas utilizam 3 operandos.
 1. Armazenar o operando "5" no registrador R1;
 2. Carregar o operando localizado no endereço 0xA4B7 da Memória Principal no registrador R2;
 3. Somar os operandos localizados em R1 e R2 e armazenar o resultado em R3;
 4. Armazenar o operando localizado em R3 no endereço 0x87FE da Memória Principal;
 5. Carregar para R4 o operando da Memória Principal localizado no endereço contido no endereço armazenado em R7;
 6. Carregar para R5 o operando localizado no endereço da Memória Principal contido em R6;
 7. Subtrair os operandos localizados em R4 e R5 e armazene o resultado em R8;
 8. Armazenar o operando localizado em R8 no endereço 0xA451 da Memória Principal.
2. Determine e descreva o modo de endereçamento utilizado em cada uma das instruções que compõem o segmento de código gerado no exercício 1.
3. Gere um segmento de código para um processador MIPS que desvie o fluxo de execução para um label denominado Exit caso uma variável 'a', armazenada em \$s0, for menor do que uma variável 'b', armazenada em \$s1.
4. Supondo que o endereço base de um array de words tenha sido armazenado previamente no registrador \$s3, gere um trecho de código em assembly, para a arquitetura MIPS, que represente a seguinte expressão:

$$A[12] = h + A[8]$$

5. O endereço base do array A de 100 words foi previamente armazenado em \$s3 e as variáveis g, h e i foram associadas aos registradores \$s1, \$s2 e \$s4 de um processador MIPS. Qual deve ser o código assembly para a operação $g = h + A[i]$ utilizando apenas as instruções lw e add?
6. Apresente o código assembly, para a arquitetura MIPS, da função swap abaixo supondo que: xp em \$a0 e yp em \$a1.

```
void swap(int *xp, int *yp)
{
    int t0 = *xp;
    int t1 = *yp;
    *xp = t1;
    *yp = t0;
}
```