



**UFES – Universidade Federal da Fronteira Sul**  
**Curso de Ciencia da Computação**  
**Disciplina – Organização de Computadores**  
**Exercícios Assembly**

- 1) O programa abaixo tenta copiar palavras do endereço apontado pelo registrador s0 para o endereço apontado pelo registrador s1, contando o número de palavras copiadas no registrador a0. O programa para de copiar quando ele encontra uma palavra com valor igual a 0 (zero), que não é incluída na contagem. Existem vários erros nesse código, arrume-os e mostre uma versão sem erros

```
loop:
    lw t1, 0(s0)           # le a proxima word da fonte
    addi a0, a0, 1         # incrementa o contador de words copiadas
    sw t1, 0(a1)           # escreve no destino
    addi a0, s0, 1         # avança para a proxima word na fonte
    addi s1, a1, 1         # avança para a proxima word no destino
    bne s0, zero, loop     # se copiado != 0, volta para loop
```

- 2) Implemente um programa que leia 10 valores (utilizando chamadas de sistema) e armazene-os em um vetor. O programa deve apresentar na saída padrão (utilizando chamadas de sistema):
- a) O maior valor lido e a sua posição dentro do vetor;
  - b) O menor valor lido e sua posição dentro do vetor;
  - c) A diferença entre o menor e o maior valor;
  - d) A quantidade de elementos com valor maior que a diferença (letra c)
  - e) A quantidade de elementos com valor menor que a diferença (letra c)
- 3) Sabendo que o rótulo inicio foi armazenado no endereço de memória 0x50000 e considerando o formato das instruções do processador RISC-V, qual o código de máquina (binário) deste programa?

```
                addi s0, zero, 1
inicio:
                beq t1, zero, fim
                addi s6, a0, -4
                add  t6, zero, t0
                lw   t1, -20(s6)
                j    inicio
fim:            add zero, zero, zero
```

- 4) Implemente uma função utilizando o conjunto de instruções do RISC-V que recebe como argumento os endereços iniciais de três vetores (A, B e C) e o tamanho dos mesmos (são iguais em tamanho) nos registradores a0 à a3. Para cada elemento a função verifica se o valor presente em A é maior que o valor presente em B, caso sim

ele coloca 1 no vetor C, se forem iguais coloca 0 no vetor C e se for menor coloca -1 no vetor C. Exemplo:

A = [4, -5, 2, 6, -2, 0]

B = [4, 6, -5, 8, -3, 2]

resultado

C = [0, -1, 1, -1, 1, -1]

- 5) Dado o código de máquina do programa implementado no RISC-V converta-o para assembly

```
<main>:
    000001100100000000000010010010011
    0000000001010000000000001010010011
    0000000000000000000000010000110011
<condicao>:
    000000000101010000000100001100011
    000000000100001001000010010110011
    00000000000101000000010000010011
    111111110101111111111000001101111
<fim>:
    0000000000000000000000000000010011
```

- 6) Escreva uma função recursiva utilizando o assembly do RISC-V que recebe um número em a0 e calcula o fatorial deste número. O resultado é retornado no registrador a0. Dica: utilize a pilha para as chamadas recursivas.
- 7) Implemente uma função que calcula o e-nesimo elemento de Fibonacci. A função recebe o elemento a ser calculado em a0 e retorna o valor correspondente também em a0. Os 6 primeiros elementos de Fibonacci são: 1, 1, 2, 3, 5, 8...