

Universidade Federal do Rio Grande do Norte
IMD0041 – Introdução a Organização e Arquitetura de Computadores
Lista de Exercícios

QUESTÃO 1 - Descreva, em pseudo-código ou em linguagem C, o comportamento do Código 1.

```
.data
max1: .word 40
min1: .word 20
.text

.globl main

main:
    li $v0, 5
    syscall
    move $t0, $v0

    lw $s0, max1
    lw $s2, min1
    li $s1, 0
    li $t1, 1
    slt $t2, $t0, $s2
    beq $t2, $t1, FORA
    slt $t3, $s0, $t0
    beq $t3, $t1, FORA
DENTRO:
    li $s1, 1
    j EXIT
FORA:
    li $s1, 0
    j EXIT
EXIT:
    li $v0, 1
    move $a0, $s1
    syscall

    li $v0, 10
    syscall
```

Código 1

QUESTÃO 2 - Implemente em assembly MIPS o Código 2, abaixo. Nesse caso, use declarações e operações de **memória** para que as variáveis k, f, g, h, i e j, estejam sempre atualizadas com seus valores correntes em **memória**.

```
int main() {
    int k, f, g=5, h=7, i=56, j=9;
    printf("Digite o valor de k\n");
    scanf("%d", &k);
    switch (k) {
        case 0: f = i + j; break;
        case 1: f = g + h; break;
        case 2: f = g - h; break;
        case 3: f = i - h; break;
    }
    printf("f= %d", f);
    return 0;
}
```

Código 2

QUESTÃO 3: Implemente o algoritmo de cálculo de fatorial em assembly MIPS. Use, nesse caso, a instrução de multiplicação disponível no simulador MARS.

QUESTÃO 4: Verifique o comportamento da unidade de controle do simulador MARS (podemos verificar a unidade de controle funcionando usando o MIPS X-Ray, usando o menu tools > MIPS X-Ray) para as instruções: addi, add, lw, sw e j. Analise cada uma das instruções em função do caminho de dados (o caminho destacado pelo MIPS X-Ray) e descreva o que precisa ser feito, pela unidade de controle, para executar corretamente cada uma das instruções.