

---

# ARQUITETURA DE COMPUTADORES

---

## Prática 3

Resolva os seguintes exercícios, escrevendo a solução usando RISC-V assembly. Para auxiliar a resolução dos exercícios utilize os guias de consulta rápida disponibilizados na página da UC.

1. Escreva o código assembly correspondente ao código C apresentado. Considere que as variáveis `x` e `y` se encontram nos registos `x10` e `x11`.

```
if (x > 10)
    if (y < 1)
        x -= 10;
    else
        x += 10;
```

2. Escreva o código que determina o número de bits a um numa variável de 32 bits em memória.
3. Escreva o código que soma `N` valores de um conjunto de valores que se encontram em memória. Por exemplo,

```
n_valores:    .word 8
valores:      .word 1,2,3,4,...
```

4. Escreva o código que troca o valor de dois registos utilizando a pilha.
5. Considere as seguintes funções em C:

```
// retorna a soma dos dois argumentos
// (passagem de parametros por registo)
int soma(int a, int b) {
    return a + b;
}

// retorna a soma dos três argumentos
// (passagem de parametros pela pilha)
int soma3(int a, int b, int c) {
    return soma(soma(a,b),c);
}

// retorna a diferença dos argumentos
// (passagem de parametros pela pilha)
int subtrai(int a, int b) {
    return a - b;
}
```

- a) Escreva o código assembly que realiza as seguintes operações utilizando estas rotinas:

```
subtrai(soma3(4, 7, 7), 3) // 4+7+7-3
```

- b) Escreva o código das rotinas.

- c) Considere o seguinte código:

```
soma3(4, 7, 7) // 4+7+7
```

Indique o conteúdo da pilha do processador quando esta está no seu máximo e à entrada e saída de cada função.

6. Considere duas rotinas designadas por `le_elemento_matriz` e `guarda_elemento_matriz`, cujo cabeçalho em código C se apresenta de seguida.

```
// Retorna o valor da linha i e coluna j da matriz.
// A matrix tem n_colunas colunas
int le_elemento_matriz(int endereco, int i, int j, int n_colunas);

// Guarda valor na linha i coluna j da matriz.
// A matrix tem n_colunas colunas
void guarda_elemento_matriz(
    int valor, int endereco, int i, int j, int n_colunas);
```

Escreva o código destas rotinas.

7. Escreva o código da rotina fatorial, implementada de forma recursiva, tal como ilustrado de seguida:

```
int factorial(int i) {
    if (i == 0)
        return 1;
    else
        return i * factorial(i - 1);
}
```

8. Escreva a rotina correspondente ao seguinte código em C que calcula  $y = x^n$ , em que a multiplicação é calculada utilizando a rotina `multiplica`.

```
int power(int x, int n) {
    int i, y = 1;
    for (i = 0; i < n; i++) {
        y = multiplica(y, x);
    }
    return y;
}
```