

Arquitectura de Computadores I

Exercícios

Licenciatura em Engenharia Informática

Aula Prática #8 (Funções, passagem de argumentos e retorno)

Em todas as questões, o seu programa deve também conter uma função **main** com a responsabilidade de preparar os argumentos e fazer as chamadas das funções. Active no simulador, no menu **Settings**, a opção **Initialize Program Counter to global 'main' if defined**.

Os programas devem ter o seguinte esqueleto:

```
.globl main
.text
main:
    # definicao de argumentos e chamadas das funcoes
```

1. Escreva uma função **max(a, b)** que calcule o maior de 2 números inteiros com sinal.
2. Escreva uma função **abs(x)** que calcule o valor absoluto de um número inteiro **x**.
3. Reescreva a função **abs(x)** de maneira a usar a função máximo. Isto é, **abs(x) = max(x, -x)**.
4. No seguinte código da função **main**, pretende-se determinar o maior de três números. Escreva a função **max3** reutilizando a função **max**, isto é, **max3(x, y, z) = max(max(x, y), z)**.

```
main:
    li a0, 3
    li a1, 5
    li a2, 2
    jal max3 # maximo de 3 numeros
```

5. Traduza a seguinte função de C para Assembly RISC-V:

```
int soma_array(int a[], int sz)
{
    int soma=0, i;

    for (i = 0; i < sz; i++)
        soma += a[i];

    return soma;
}
```

6. Escreva uma função **array_max(int a[], int sz)** que obtenha o máximo de um array não vazio de inteiros. Em cada iteração do ciclo, deve chamar a função **max()** para obter o maior entre o valor lido do array e o maior encontrado até ao momento. O código seguinte mostra o que se pretende em C:

```
int array_max(int a[], int sz)
{
    int m = a[0];
```

```
    for (i = 1; i < sz; i++)  
        m = max(m, a[i]);  
  
    return m;  
}
```