



UFAM

**Universidade Federal do Amazonas
Instituto de Computação**

**Organização de Computadores
Lista de Exercícios - Programação ARM**

Observações:

- a)Esta lista é individual**
- b)Plágio não será tolerado**

- 1- Faça um programa que defina 2 variáveis na memória e guarde em R0 o maior entre eles.
- 2- Faça um programa que armazene em R0 o resultado da soma entre os 100 primeiros números.
- 3- Faça um programa em assembly ARM que checa se um valor é par, se a condição for verdadeira o valor de R0 deve ser igual a: número * número + 2. Senão o valor de R0 deve ser igual a: (número + 1) * 2.

4- Faça um trecho de código em linguagem de montagem do ARM equivalente ao código em C abaixo, a instrução return deve ser realizada com MOV LR, RD:

```
int menor (int a, int b)
{
    if (a <= b)
        return (a);
    else
        return (b);
}
```

5- Faça um trecho de código equivalente em linguagem de montagem do ARM, tome cuidado com os ponteiros, a instrução return deve ser realizada com MOV LR, RD.

```
int soma_salarios (int *salarios, int numero_funcionarios)
{
    int total;
    int *p;

    for (p = salarios; p < salarios+ numero_funcionarios; p++)
        total = total + *p;

    return (total);
}
```

6- Faça um programa em assembly ARM equivalente ao código em C apresentado abaixo:

```
void function1 ()
{
    int z;
    int array[1024]; // local variable

    for (i = 0; i < 1024; i++)
        array[i] = i;

    z = sum(array, 1024);
    print_int(z);
}

int sum(int array[], int n)
{
    int s = 0;
    int i;

    for (i = 0; i < n; i++)
        s = s + array[i];

    return s;
}
```

7- Faça um programa em assembly ARM equivalente ao código em C apresentado abaixo:

```
int array[1024]; // variable
global

int function1()
{
    int z;
    int i;

    for (i = 0; i < 1024; i++)
        array[i] = i;

    z = sum(1024);
    print_int(z);
}

int sum(int n)
{
    int s = 0;
    int i;

    for (i = 0; i < n; i++)
        s = s + vector[i];

    return s;
}
```

8- Escreva um programa em linguagem de montagem do ARM para calcular o produto escalar de dois vetores de inteiros com 6 valores armazenados, localizados na memória.

Parâmetros de entrada: R0 = endereço do 1º vetor e R1= endereço do 2º vetor.

Parâmetro de saída: (LR)= produto escalar.

9- Escreva um programa que calcula os 10 primeiros elementos da série de fibonacci. Use a instrução `Store` para guardar na posição de memória referente a iteração o elemento da série.

10- Faça um trecho de código em assembly ARM para fazer a multiplicação de dois números de 16 bits sem sinal sem usar a instrução mult. Suponha que os valores a serem multiplicados já estão guardados em dois registradores do processador e indique em que registrador o resultado da multiplicação estará guardado após a execução do trecho de código (escolha livremente estes registradores). Não use pseudo instruções e inclua um comentário para cada instrução.

11- Faça um trecho de código em assembly ARM para fazer a divisão de dois números de 16 bits sem sinal sem usar a instrução div. Suponha que os valores a serem divididos já estão guardados em dois registradores do processador e indique em que registrador(es) o resultado da divisão (o quociente e o resto) estará guardado após a execução do trecho de código (escolha livremente estes registradores). Não use pseudo instruções e inclua um comentário para cada instrução.

12- Faça em assembly ARM o bubble sort apresentado abaixo em pseudocódigo:

Algoritmo Bubble

variáveis

inteiro: aux, num[7] := { 7, 5, 2, 1, 1, 3, 4} ,i, j,

MAX;

início.

MAX := 7;

para i de 0 até (MAX-1) faça

para j de (MAX-1) até (i+1) passo -1 faça

se num[j-1] > num[j] então

aux := num[j-1];

num[j-1] := num[j];

num[j] := aux;

fim_se;

fim_para;

fim_para;

para i de 0 até MAX faça

escreva(num[i]);

fim_para;

fim.