



**UNIVERSIDADE FEDERAL DA PARAÍBA**  
**CENTRO DE INFORMÁTICA**  
**DISCIPLINA: Arquitetura de Computadores**  
**Professor: Ewerton Salvador**

**LISTA DE EXERCÍCIOS – ASSEMBLY**

Tente implementar programas em Assembly para resolver os problemas abaixo, considerando as seguintes observações:

- Desenvolva os programas para funcionar na console do sistema operacional;
- Para evitarmos, nesse primeiro momento, complicações que podem surgir decorrentes da implementação de funções para entrada de dados, faça com que todos os dados de entrada que o seu programa necessitar sejam disponibilizados através de valores de inicialização de variáveis. Você pode inicializar essas variáveis da forma que desejar;
- Para realizar saída de dados, utilize uma forma simples de escrita na console, como a macro `printf` no MASM32 ou a função `printf` no NASM com programa “linkado” com a biblioteca de entrada e saída da linguagem C;
- Lembre-se que algumas funções poderão apresentar “efeitos colaterais”, ou seja, elas podem causar alterações no estado do computador (alterar valores de registradores ou de posições de memória) sem que isso esteja explícito na chamada da função. Por exemplo, a macro “`printf`” do MASM32 altera os valores dos registradores EAX, ECX e EDX após a sua execução. Se os valores desses registradores precisam ser preservados, você pode salvar esses registradores na pilha (ou seja, fazer um backup na memória) antes de chamar a macro `printf`, e recuperar os valores originais desses registradores após a chamada do `printf`;
- Cuidado com nomes de variáveis e de rótulos (*labels*) muito comuns – esses nomes precisam ser únicos por todo o programa, e sempre existe a chance do nome já ter sido utilizado em alguma das bibliotecas utilizadas! Considere utilizar nomes representativos, mas que possuam baixa probabilidade de coincidirem com nomes já utilizados em bibliotecas que você usará.

1. Escreva um programa que calcule a soma dos 100 primeiros números inteiros positivos. O resultado deverá ser armazenado no registrador *eax* e também deverá ser exibido na tela.

2. Escreva um programa que implemente a seguinte sentença da linguagem Java:

`a = b + c + 100;`

As variáveis *a*, *b* e *c* são valores inteiros armazenados na memória. O conteúdo das variáveis *b* e *c* deverão ser inicializados com valores definidos por você.

3. Escreva um programa que leia duas constantes numéricas inteiras e imprima o maior dentre os dois números informados. Se os valores forem iguais, o programa pode imprimir qualquer uma das variáveis.

4. Escreva um programa que leia uma constante numérica inteira e, em seguida, escreva na tela se o número é par ou ímpar.

5. Escreva um programa que exiba na console os números entre 1000 e 1999 que divididos por 11 dão resto 5.

6. Escreva um programa que receba um array de 10 números inteiros **de dois bytes (tipo WORD)** e que exiba o valor da soma desses 10 números.