



UNIVERSIDADE FEDERAL DO CEARÁ  
Campus de Quixadá  
Prof. Thiago Werlley Bandeira da Silva  
QXD0132- Arquitetura e Organização de Computadores I

# Trabalho

## 2024.1

Nome: \_\_\_\_\_ Matrícula: \_\_\_\_\_

### 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;
  - 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 números par e ímpar de 1 até 100. 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:

$$a = (2 * b) + (c/2) + 200;$$

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 10 constantes numéricas inteiras e imprima o menor e o maior dentre os números informados.
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.
7. Escreva um programa que receba um array de 5 números inteiros de dois bytes (tipo `WORD`) e que exiba o valor da multiplicação desses 5 números.
8. Escreva um programa que apresente 10 valores em um vetor e imprima o segundo menor e o segundo maior dentre os números informados.
9. Escreva um programa que calcule a média ponderada das 3 notas de um aluno. A primeira nota tem peso 1, a segunda tem peso 2 e a terceira tem peso 3.

$$\text{Media} = ((N_1 * 1) + (N_2 * 2) + (N_3 * 3))/3;$$

Imprima a situação do aluno dependendo da sua média final:

- “Aluno Aprovado” (para média maior ou igual a 7)

Nota: \_\_\_\_\_

- “Aluno em Avaliação Final” (para média maior ou igual a 4 e menor que 7)
- “Aluno Reprovado” (para média menor que 4).

10. Converta o programa em C para NASM 32 bits.

```
1 #include <stdio.h>
2
3 int main(void) {
4     int media = 0;
5     while (media != -1) {
6         printf("Digite a media do final aluno: \n");
7         scanf("%d", &media);
8         if (media < -1 || media > 10) {
9             printf("Voce digitou numero invalido ");
10            continue;
11        }
12        printf(media >= 7 ? "APROVADO!\n" : "REPROVADO!\n");
13    }
14    printf("Fim do programa! \n"); // quando eu digito -1
15 }
```