

3º Laboratório de Programação para Sistemas Embarcados

Prof. Otávio Gomes e Prof. Rodrigo Almeida

Leia com atenção - Informações iniciais:

1. No início de cada tópico/assunto é apresentado um **exercício de revisão** em que basta copiar o código na ferramenta, realizar a compilação e a execução e, então, interpretar o resultado. Este tipo de exercício tem como objetivo auxiliar o aluno a relembrar alguns conceitos e a validar as ferramentas que estão sendo utilizadas. Este código sempre estará correto e funcionando.
2. Os exercícios estão apresentados em **ordem crescente de dificuldade**.
3. Para o **registro de frequência**, o aluno deverá enviar o código relativo ao exercício mais difícil desta lista que conseguir resolver. Por exemplo:
 - a. Se em uma lista contendo 6 exercícios o aluno A conseguiu resolver até o exercício 4, é este que ele deve enviar para registro de frequência.
 - b. Se o aluno B conseguiu resolver toda a lista de exercícios, deve enviar o último exercício da lista.
4. Os exercícios abordam todos os conceitos relacionados ao conteúdo da aula em questão. Deste modo, caso o aluno não consiga resolver alguns dos exercícios, recomenda-se que o mesmo participe dos **plantões de dúvidas** e que busque aprender os conceitos envolvidos na atividade.
5. A **próxima atividade** de laboratório admitirá que os conceitos aqui apresentados já foram plenamente compreendidos.
6. A **entrega** desta atividade para o controle de frequência será realizada pelo SIGAA.

1) A aplicação a seguir foi utilizada no exercício 1 do roteiro anterior. Agora, neste roteiro, a mesma aplicação foi modificada com a utilização de funções:

```
#include <stdio.h>
#include <stdlib.h>

int max = 0;
int min = 0;
int pares = 0;

void inicializa_variaveis(int val);
int retorna_max(int val);
int retorna_min(int val);
void conta_pares(int val);

int main(){
    int pos = 10;
    int vet[pos];
    int valor=0;
    for(int i=0; i<=(pos-1); i++){
        printf("Forneca um valor inteiro positivo (elemento %d/%d): ", i+1, pos);
        scanf("%d", &valor);
        if(valor<0){ valor = 0; }
```

```
vet[i] = valor;
if(i==0){
    inicializa_variaveis(valor);
}
else{
    max = retorna_max(valor);
    min = retorna_min(valor);
}
conta_pares(valor);
}

printf("\nOs elementos do vetor são: ");
for(int i=0; i<=(pos-1); i++){
    printf("%d ", vet[i]);
}

printf("\nMaior elemento: %d", max);
printf("\nMenor elemento: %d", min);
printf("\nNúmero de elementos pares: %d", pares);

return 0;
}

void inicializa_variaveis(int val){
    max = val;
    min = val;
    pares = 0;
}

int retorna_max(int val){
    if (val>max){ return valor; }
    else return max;
}

int retorna_min(int val){
    if (val<min){ return valor; }
    else return max;
}

void conta_pares(int val){
    if (val%2 ==0){ pares++; }
}
```

- Que modificações foram feitas no código?
- De que modo a utilização de funções auxilia no desenvolvimento e manutenção do código?

- 2) Crie uma função que receba como parâmetro um número inteiro e devolve o seu dobro.
- 3) Elabore uma função que receba três notas de um aluno como parâmetros e uma letra. Se a letra for A, a função deverá calcular a média aritmética das notas do aluno; se for P, deverá calcular a média ponderada, com pesos 5, 3 e 2.
- 4) Escreva uma função que receba um número inteiro maior do que zero e retorne a soma de todos os seus algarismos. Por exemplo, ao número 251 corresponderá o valor 8 ($2 + 5 + 1$). Se o número lido não for maior do que zero, o programa terminará com a mensagem “Número inválido”.
- 5) Escreva uma função para determinar a quantidade de números primos abaixo N. O número N será fornecido como parâmetro para a função.
- 6) Crie um programa contendo as seguintes funções que recebem um vetor V números reais como parâmetro:
 - a) Impressão normal do vetor.
 - b) Impressão inversa.
 - c) Função que retorna a média aritmética dos elementos do vetor.