



Lista 2 – Computação Científica A

Estruturas Homogêneas:

1. Desenvolver um programa para ler um vetor V de elementos inteiros de tamanho N ($N \leq 50$). Separar este vetor em dois vetores A e B . O vetor A deve conter os elementos positivos de V e o vetor B os elementos negativos de V . Os elementos nulos de V não devem ser gravados, mas devem ser contados. Escrever os vetores V , A e B . Escrever quantos elementos nulos foi encontrado em V .
2. Faça um programa que leia 200 valores do tipo inteiro e os guarde na memória. Determine e mostre qual o menor valor existente no vetor, qual o maior valor existente no vetor. Também, calcule e mostre quantos dos valores armazenados no vetor são maiores que a média desses valores.
3. Dado nome e salário de 10 funcionários faça um programa que leia estes valores e:
 - a) Imprima o nome e o salário dos funcionários que ganham R\$ 500,00 ou mais;
 - b) Para os funcionários que ganham menos de R\$ 200,00 conceda um aumento de 20%. Imprima o nome e o novo salário destes funcionários.
4. Ler uma matriz A ($M \times N$) de elementos inteiros, com M e $N \leq 10$. A seguir, multiplicar a matriz por um valor inteiro (multiplicação por escalar) e imprimir a matriz resultante no formato tabular.
5. Ler uma matriz quadrada A de ordem M ($M \leq 10$) de elementos inteiros e verificar se a mesma é triangular inferior. Nota: uma matriz é triangular superior quando todos os seus elementos situados abaixo da diagonal principal são iguais a zero.
6. Ler uma matriz A ($M \times N$) de elementos inteiros, com M e $N \leq 10$, e determinar a coluna que possui a maior soma de elementos. O total e os valores desta coluna devem ser exibidos. Em caso de empate, mostrar a coluna que aparece primeiro.
7. Ler uma matriz quadrada de ordem 2, calcular e mostrar uma matriz resultante que será a matriz digitada multiplicada pelo maior elemento da matriz.
8. Ler uma matriz quadrada de ordem 10, calcular e mostrar o traço da matriz. Chama-se traço da matriz a soma dos elementos contidos na diagonal principal da matriz.

Funções:

1. Faça uma função que calcule a enésima potência de uma variável real x :

$$f(x, n) = x^n.$$

2. Faça uma função que calcula o fatorial de um número. Implemente a versão recursiva (sem laço de repetição).

3. Dado que podemos calcular

$$e^x = 1 + x + x^2/2! + x^3/3! + \dots$$

Faça um programa utilizando funções que leia o valor de x , e calcule o valor de e^x enquanto o termo $x^n/n! > 10^{-6}$.

4. Faça um programa em "C", que solicita o gasto total de um cliente e a condição de pagamento desejada, podendo ser uma das opções abaixo:

Opção 1: a vista (10 % de desconto)

Opção 2: em duas vezes (preço da etiqueta)

Opção 3: de 3 até 10 vezes (3% de juros ao mês)

OBS: deverão ser feitas três funções, uma para cada condição de pagamento. Assim, cada função deverá apresentar as condições de pagamentos, apresentando a(s) parcela(s) mensal/(ais) e o valor total pago.

Ponteiros:

1. Procure determinar quais valores são impressos ao final deste programa. Confira sua resposta testando o programa no micro. Execute-o passo a passo conferindo o valor das variáveis em cada momento.

a) void main()

```
{
    int a,b,*c;

    a = 3;
    b = 4;
    c = &a;
    b++;
    *c = a+2;
    printf("%d %d",a,b);
}
```

b) void main()

```
{
    int a,b,*c;

    a = 4;
    b = 3;
    c = &a;
    *c = *c +1;
    c = &b;
    b = b+4;
    printf("%d %d %d",a,b,*c);
}
```

c) void main()

```
{
    int a,b,*c,*d,*f;

    a = 4;
    b = 3;
    c = &a;
    d = &b;
    *c /= 2;
    f = c;
    c = d;
    d = f;
    printf("%d %d",*c,*d);
}
```

d)

int calcula(int);

void main()

```
{
```

```

int a,b,c;
char d;
a=1;b=2;c=3;d='A';
a+=b*c;
d=(a>7)?d-1:d+1;
b = calcula(b);
c = calcula(calcula(a));
a = c++;
printf("%d - %d - %d - %c\n",a,b,c,d);
}

int calcula(int x)
{
    int i;

    if ((x=x*2)>5) return(x+3);
    for(i=0;i<10;i++)
    {
        if (i<5) continue;
        if (x>8) break;
        x+=2;
    }
    return(x);
}

```

Arquivos:

1. Escreva um programa que:

- (a) Crie/abra um arquivo texto de nome "arq.txt"
- (b) Permita que o usuário grave diversos caracteres nesse arquivo, até que o usuário entre com o caractere '0'
- (c) Feche o arquivo

Agora, abra e leia o arquivo, caractere por caractere, e escreva na tela todos os caracteres armazenados.

- 2. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas linhas esse arquivo possui.
- 3. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas letras são vogais.
- 4. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas letras são vogais e quantas são consoantes.
- 5. Faça um programa que receba do usuário um arquivo texto e um caractere. Mostre na tela quantas vezes aquele caractere ocorre dentro do arquivo.
- 6. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas vezes cada letra do alfabeto aparece dentro do arquivo.
- 7. Faça um programa que receba do usuário um arquivo texto. Crie outro arquivo texto contendo o texto do arquivo de entrada, mas com as vogais substituídas por '*'.
- 8. Faça um programa que leia o conteúdo de um arquivo e crie um arquivo com o mesmo conteúdo, mas com todas as letras minúsculas convertidas para maiúsculas. Os nomes dos arquivos serão fornecidos, via teclado, pelo usuário. A função que converte maiúscula para minúscula é o `toupper()`. Ela é aplicada em cada caractere da string.
- 9. Faça um programa que receba dois arquivos do usuário, e crie um terceiro arquivo com o conteúdo dos dois primeiros juntos (o conteúdo do primeiro seguido do conteúdo do segundo).
- 10. Faça um programa que receba o nome de um arquivo de entrada e outro de saída. O arquivo de entrada contém em cada linha o nome de uma cidade (ocupando 40 caracteres) e o seu número de habitantes. O programa deverá ler o arquivo de entrada e gerar um arquivo de saída onde aparece o nome da cidade mais populosa seguida pelo seu número de habitantes.
- 11. Faça um programa no qual o usuário informa o nome do arquivo e uma palavra, e retorne o número de vezes que aquela palavra aparece no arquivo.
- 12. Você foi contratado por uma empresa de transporte logístico. A empresa utiliza a seguinte base de dados para cadastro de motoristas e veículos, salvos em um arquivo de texto:
cadastro.txt

Como você verificou, o arquivo de texto possui uma máscara formatada do tipo “%s\t%s\t%d\n”. Neste caso simplificado, considere que os motoristas são cadastrados apenas pelo primeiro nome, sendo que não existirão motoristas com o mesmo nome. Considere também que o arquivo de texto já existe. Sua função é criar um algoritmo em C para cadastro de motoristas e veículos.

- a. Crie uma função que recebe o nome do arquivo de texto, e os dados do registro (motorista, veículo e ano), esta deverá adicionar o registro sempre no final do arquivo de textos.
- b. Crie uma função para verificar se o motorista que se deseja cadastrar já existe no registro. Caso exista a função deverá retornar erro informando que o motorista já possui um carro associado.
- c. Crie o corpo principal do programa em C que utiliza as funções criadas nos itens (a) e (b).

13. Considerando o exercício anterior desenvolva a funcionalidade de atualizar o veículo do motorista. Para simplificar, você poderá criar um novo arquivo de textos com os registros atualizados.

***Dicas:

- Elabore algoritmos narrativos, fluxogramas e pseudocódigos antes de programar o código dos exercícios.
- Sempre que possível tente programar em linguagem C primeiro no papel e depois no computador.