



Faculdade de Computação

Programação Procedimental

4º Laboratório de Programação C

Prof. Cláudio C. Rodrigues

1. Introdução

O objetivo desta aula prática é explorar os agregados homogêneos (**arrays**, **strings**).

Vetores, também conhecidos como *arrays*. Vetores são estruturas homogêneas que servem para guardar vários valores do mesmo tipo de forma uniforme na memória. Por exemplo, se for necessário criar 20 variáveis do mesmo tipo e que tenham o mesmo significado, nós não criaríamos \Rightarrow `int x1, x2, x3, x4, x5 ...`, no lugar disso, criaríamos apenas uma variável não escalar denominada **vetor** para guardar todos os 20 números.

Como um vetor pode guardar vários valores temos que definir quantos valores ele deve guardar para que seja reservado o espaço necessário em memória. Então, definimos a declaração de um vetor da seguinte maneira:

<tipo> identificador [<número de elementos>];

- **tipo** de dado: `int`, `float`, `double`, ...
- **identificador**: nome da variável, usando as mesmas convenções de uma variável comum. (`array`, `vetor`, `variavelDeNumeros`, ...)
- quantidade de **elementos**: o tamanho do vetor escrito entre colchetes: `[5]`, `[10]`, `[3]`...

2. Problemas

P1) Escreva em linguagem C um programa que leia dois vetores de cinco elementos reais cada; e defina um terceiro vetor com a soma dos elementos dos dois vetores.

Dica $C[i] = A[i] + B[i]$

O algoritmo deve estar contido no arquivo "**somavet.c**".

P2) Na teoria dos sistemas define-se o elemento MINMAX de uma matriz como sendo o maior elemento da linha onde se encontra o menor elemento da matriz. Elabore um algoritmo em linguagem C que carregue uma matriz 4x5 com números reais, calcule e mostre o MINMAX e sua posição (linha e coluna).

O algoritmo deve estar contido no arquivo "**minmax.c**".

P3) Dizemos que uma matriz inteira $A(n \times n)$ é uma matriz de permutação se em cada linha e em cada coluna houver $n-1$ elementos nulos e um único elemento igual a 1. Escreva um programa em linguagem C que leia uma matriz inteira $A(n \times n)$ e verifique se A é de permutação.

Exemplos:

```
0 1 0 0
0 0 1 0
1 0 0 0
0 0 0 1
```

O algoritmo deve estar contido no arquivo "**permutacao.c**".

- P4)** Escreva um programa em linguagem C que verifique se a matriz A é simétrica. Uma matriz $A_{n \times n}$ é simétrica se, e somente se, A é igual a sua transposta, ou seja, $A = A^t$, isto é, se $A[i][j] == A[j][i]$;

Note que devemos percorrer somente uma parte da matriz, no caso, a parte triangular superior da matriz. O algoritmo deve estar contido no arquivo "**simetrica.c**".

- P5)** Um treinador de atletismo treina 5 atletas e faz 6 sessões de treino por semana. Em cada sessão, cada atleta percorre uma distância que é cronometrada. Os valores dos tempos, em segundos, são registrados sob a forma de uma matriz, onde cada linha diz respeito a um atleta e cada coluna a uma sessão de treino. Supondo já feita a leitura da matriz, escreva as linhas de código necessárias para:

- Calcular e escrever a média dos tempos realizados para cada atleta nas 6 sessões de treinos;
- Determinar e escrever o melhor tempo realizado por cada um dos atletas nas 6 sessões.

O algoritmo deve estar contido no arquivo "**treino.c**".

- P6)** Escreva em linguagem de programação C um programa que recebe como parâmetro de entrada uma string (cadeia de caracteres) e escreve na tela um histograma de *frequência* de letras vogais presentes na string recebida. Para elaborar a solução, faça todas as considerações necessárias. O programa deve declarar todas as variáveis necessárias para a realização da tarefa, bem como, caracterizar o elemento letra vogal e contabilizar da mesma forma as vogais maiúsculas e minúsculas. A função deve retornar a quantidade total de letras vogais identificadas na string de entrada. O histograma de frequência de vogais deve ser construído com o formato do exemplo abaixo.

Exemplo:

>char str[] = "O unico homem que esta isento de erros, eh aquele que nao arrisca acertar";

Histograma:

07 a *****

11 e *****

03 i ***

06 o *****

04 u ****

O algoritmo deve estar contido no arquivo "**frequencia.c**".

- P7)** Escreva um programa em linguagem C que faça a leitura e armazene num vetor de 10 posições o nome dos alunos aprovados no vestibular para Computação em 2016/1. A seguir, leia um nome de entrada e verifique se este aluno consta na listagem dos aprovados e mostre sua classificação (ordem de aprovação, representada pelo índice do vetor +1). Caso o nome informado não exista na lista dos aprovados, mostre uma mensagem "Nome informado não consta na lista dos aprovados".

Ex:

Curso: Computação	
Classificação:	Nome
1	Ana Santos
2	João Souza
3	Maria Silveira
...	Fulano de Tal
10	André Cajarana

O aluno João Souza foi aprovado em 2º. Lugar

O algoritmo deve estar contido no arquivo "**busca.c**"