

# UFU - FACOM: Algoritmos e Programação de Computadores

Professores: Ronaldo Castro de Oliveira

**3ª lista de exercícios:** em todos os exercícios abaixo, elaborar um programa em C++ que:

## Strings

- 1) Leia 3 nomes de alunos quaisquer (aluno1, aluno2 e aluno3) e imprima estes nomes em ordem alfabética.
- 2) Leia um nome de uma pessoa e imprima este nome invertido. Ex: Nome = PROGRAMA - Invertido = AMARGORP.
- 3) Leia os dados de N alunos incluindo o nome e a nota final. O programa deve calcular a média de notas dos alunos e mostrar uma lista de nomes dos alunos que tiveram notas acima da média.
- 4) Leia um conjunto de N nomes de pessoas e mostre o nome que viria primeiro (menor nome) e o nome que viria por último (maior nome) em uma ordem alfabética.
- 5) Leia uma lista de N pessoas de uma agenda incluindo NOME e TELEFONE de cada um. Após ler os dados o programa deve ordenar esta lista por NOME em ordem alfabética e imprimir a lista completa contendo o nome e o telefone das pessoas.

## Estruturas heterogêneas - Structs

6) Um programa que lê os valores do comprimento dos dois catetos de um triângulo retângulo, e uma função calcula o valor da hipotenusa, e a função main imprime este valor com uma mensagem explicativa. Use a estrutura para representar o triângulo retângulo:

```
typedef struct {
    float a;
    float b;
    float hip;
} triang_ret;
```

7) Defina uma estrutura empregado para armazenar nome, data de nascimento, idade e salário de um empregado de uma empresa. Leia informações sobre N empregados da empresa usando estruturas. Atualize o campo salário de cada empregado, aplicando um reajuste salarial, de acordo com a tabela abaixo, e em seguida imprima os dados de cada funcionário.

Valor do salário	Porcentual de reajuste
até R\$880.50	15%
acima de R\$880.50, e até R\$5000.00	10%
acima de R\$5000.00	5%

O programa devera ter as seguintes funções: **le\_empregado**, **mostra\_empregado** e **atualiza\_salario**

8) Um racional é qualquer número da forma  $p/q$ , sendo p inteiro e q inteiro não-nulo. Assim, é conveniente representar cada racional por uma estrutura:

```
typedef struct {
    int p;
    int q;
} racional;
```

Usando a esta estrutura, escreva funções que:

- receba dois inteiros x e y e devolva o racional que representa  $x/y$ ;
- receba dois racionais x e y e devolva o racional que representa a soma de x e y;
- receba dois racionais x e y e devolva o racional que representa o produto de x por y;
- receba dois racionais x e y e devolva o racional que representa o quociente de x por y;

A função principal deve ler quatro números inteiros e, a partir deles, obter dois números racionais. Em seguida deve chamar as funções e imprimir os resultados das operações de soma, multiplicação e divisão desses dois números racionais.

9) Uma Universidade deseja fazer um levantamento a respeito de seu concurso vestibular e, registrou algumas informações sobre os seus N cursos, tais como: código do curso, número de vagas, número de candidatos do sexo masculino e número de candidatos do sexo feminino. Implementar um programa em C que:

- leia as informações para cada curso, armazenando em um vetor de estrutura contendo:
  - os códigos dos cursos COD;
  - os números de vagas VAGAS;
  - os números de candidatos do sexo masculino e do sexo feminino, respectivamente, na primeira e segunda posição do vetor NUM com 3 posições.
- calcule para cada curso, o número de candidatos inscritos; estes números deverão estar armazenados na terceira posição do vetor NUM;
- calcule e imprima, para cada curso, o número de candidatos por vaga, deverá ser armazenado no vetor CAND;
- determine o maior número de candidatos por vaga e imprima este número juntamente com o código do curso correspondente (supor que não haja empate);
- dado o código de um curso qualquer, imprima o número de vagas e o número de candidatos por vaga, juntamente com o código do curso. Caso o código desejado não esteja cadastrado, imprima: 'CURSO INEXISTENTE'.

10) Fazer um programa que leia os dados de pessoas de uma agenda incluindo NOME e TELEFONE de cada um. O programa deverá usar estruturas e ter um menu as seguintes opções:

- Função **le\_pessoa** que deverá cadastrar uma pessoa no sistema;
- Função **lista\_pessoas** que deverá listar todas as pessoas cadastradas no sistema;
- Função **ordena\_pessoas** que deverá ordenar todas as pessoas cadastradas em ordem alfabética pelo nome utilizando o método de ordenação de seleção;
- Função **busca\_pessoa** que deverá ler um nome pelo teclado e se a pessoa existir mostra o telefone cadastrado desta pessoa;
- Uma função **apaga\_pessoa** que deverá ler um nome pelo teclado e se a pessoa existir mostra o nome e telefone cadastrado desta pessoa. A função deve confirmar com o usuário se quer mesmo excluir a pessoa apresentada. Se sim, a função apaga a pessoa da seguinte forma.
  - Seja o índice da pessoa encontrada como sendo i de um conjunto de N pessoas;
  - Fazer um laço da variável x de i+1 até N;
    - Copiar os dados da estrutura na posição x na posição x-1;
    - Incrementar x;
  - Decrementar N indicando que uma pessoa foi excluída;
  - Apresentar mensagem de exclusão da pessoa.

11) O sistema de avaliação de uma determinada disciplina é definido por 4 notas no valor de 25 pontos cada uma, assim distribuídas: um trabalho, prova1, prova2, e prova3. A nota final do aluno é calculada pela soma das 4 notas, número reais, que o aluno tirou no semestre. O aluno será aprovado se sua nota final for maior ou igual a 60 e o número de faltas for menor que 18.

Faça um programa em C(++) onde:

1. A função **le\_dados** que lê os seguintes dados dos alunos: o número de matrícula (caractere), nome, o código do sexo ('f' para o sexo feminino ou 'm' para o sexo masculino), as 4 notas e o número de faltas.
2. A função **calcula\_nota\_final** que calcula a nota final de cada um dos alunos (soma das notas), e armazena na estrutura no campo **nota\_final**.
3. A função **calcula\_media** que calcula a média aritmética das notas finais da turma.
4. A função **aprovados\_Fem** que calcula quantidade de alunos aprovados, do sexo feminino.
5. A função **aprovados\_Masc** que calcula quantidade de alunos aprovados, do sexo masculino.
6. A função **porcentagem\_aprovados** que calcula a porcentagem de alunos com nota final acima da nota média da turma.
7. A função **ordena\_nota** que classifica os alunos em ordem crescente de suas notas finais.
8. A função **gera\_boletim** que gera os boletim dos alunos com os dados: número de matrícula, nome, nota final, faltas e situação de aprovação (Aprovado ou Reprovado).
9. A função **main** lê o valor de N, lê os dados dos alunos usando a função **le\_dados** e calcula a nota final dos alunos usando a função **calcula\_nota\_final**. De acordo com uma opção do usuário, fornecido na execução da função **menu**, ativa a execução da outra função (itens 3 a 8) imprimindo os resultados obtidos pela função.
10. A função **menu** imprime o menu e lê a opção do usuário:

Menu de opções:

- 1 - Nota média da turma.
  - 2 - Número de alunos, do sexo feminino, aprovados.
  - 3 - Número de alunos, do sexo masculino, aprovados.
  - 4 - Porcentagem de alunos com nota final acima da nota média.
  - 5 - Classificar os alunos, pela nota final.
  - 6 - Gerar os boletins.
  - 7 - Sair do programa.
- Qual a sua opção?

Obs. Use um vetor de estruturas para armazenar os dados dos N alunos, considerem  $N \leq 10$ .