

## Lista 2 - Exercícios Struct / Arquivos - SCC0801

### 1. Ranking de bandas preferidas:

- a. Defina uma estrutura que irá representar bandas de música. Essa estrutura deve ter o nome da banda, que tipo de música ela toca, o número de integrantes e em que posição do ranking essa banda está dentre as suas 5 bandas favoritas.
- b. Crie um looping para preencher as 5 estruturas de bandas criadas no exemplo passado. Após criar e preencher, exiba todas as informações das bandas/estruturas. Não se esqueça de usar o operador -> para preencher os membros das structs.
- c. Crie uma função que peça ao usuário um número de 1 até 5. Em seguida, seu programa deve exibir informações da banda cuja posição no seu ranking é a que foi solicitada pelo usuário.
- d. Crie uma função em C que peça ao usuário um tipo de música e exiba as bandas com esse tipo de música no seu ranking. Que função da string.h você usaria para comparar as strings que representam o tipo de banda?
- e. Crie uma função que peça o nome de uma banda ao usuário e diga se ela está entre suas bandas favoritas ou não.
- f. Agora junte tudo e crie uma mega aplicação em que exibe um menu com as opções de preencher as estruturas e todas as opções das questões passadas.
- g. Faça uma função que salve todos os dados em um arquivo .txt. Abra o arquivo e confira os dados.

### 2. Tipos de arquivos:

- a. Faça um programa que crie um arquivo TEXTO em disco, com o nome "dados.txt", e escreva neste arquivo em disco uma contagem que vá de 1 até 100, com um número em cada linha. Abra este arquivo em um editor de textos, como por exemplo o Notepad ou o Wordpad do Windows.
- b. Faça um programa que crie um arquivo BINÁRIO em disco, com o nome "dados.bin", e escreva neste arquivo em disco uma contagem que vá de 1 até 100, com um número em cada linha. Abra este arquivo em um editor de textos e observe como ficou o seu conteúdo (ilegível!).
- c. Faça 2 programas, um que leia o arquivo TEXTO criado no exercício 1 e outro que leia o arquivo BINÁRIO criado no exercício 2. Exibir na tela os dados lidos dos respectivos arquivos.

3. Crie uma estrutura de dados:
  - a. Para horarios, composta de horas, minutos e segundos.
  - b. Data, composta de dia, mês ano.
  - c. Compromissos, composta de um campo de descrição (string), horario (estrutura 1) e data (estrutura 2). Dica: Como já descrito, utiliza as estruturas anteriores de horário e data dentro dessa estrutura com um campo adicional de descrição do compromisso.
4. Faça um programa em C:
  - a. Para preencher os campos da struct do exercício anterior e outro para imprimi-los na tela.
  - b. Que permita o usuário registrar N compromissos (vetor de struct).
  - c. Que o usuário digite uma data e o programa exiba na tela todos os compromissos desta data.
  - d. Salve em um arquivo todos os compromissos de uma determinada data especificada pelo usuário.
  - e. Leia todos os compromissos de um arquivo, salve em memória e exiba na tela.
5. Gerenciamento de alunos e notas:
  - a. Faça um programa que leia e armazene em um registro de dados as seguintes informações: Nome do Aluno, Nro. USP, Nota1, Nota2 e Média. Os dados devem ser digitados pelo usuário, armazenados na estrutura e exibidos na tela (leitura dos dados de apenas 1 aluno).
  - b. Baseado no programa anterior de leitura dos dados de um aluno, faça um programa que armazene em memória os dados de toda uma turma, composta de até 5 alunos. Os dados devem ser armazenados em um vetor de estruturas compostas (nome, nro, n1, n2 e média). O programa deve ler os dados dos alunos (sem a média), armazenar no vetor, calcular a média simples das notas, e depois exibir na tela todos os dados de cada um dos alunos.
  - c. Baseado no programa anterior, faça um programa que leia, armazene e exiba os dados de toda uma turma, composta de até 50 alunos. A cada novo aluno cujos dados são informados pelo usuário, pergunte se o usuário deseja continuar a entrar novos dados, caso ele diga que sim, continue lendo os dados dos alunos até ter lido no máximo 50 dados de alunos.
  - d. Faça um programa que leia os dados de 50 alunos (ou até que o usuário diga que não deseja mais entrar dados), onde cada aluno possui as seguintes informações: Nome, Nro. USP, Nota1, Nota2. Calcule a média dos alunos usando uma média ponderada, onde a Nota1 tem peso 1 e a Nota2 tem peso 2. Armazene estes dados em um vetor em memória. Salve os dados lidos em um arquivo texto.
  - e. Faça um programa que leia os dados de 10 alunos (Nome, Nro. USP, Média Final), armazenando em um vetor. Uma vez lidos os dados, divida estes dados em 2 novos

vetores, o vetor dos aprovados e o vetor dos reprovados, considerando a média mínima para a aprovação como sendo 5.0. Exibir na tela os dados do vetor de aprovados, seguido dos dados do vetor de reprovados.