

**Universidade de São Paulo**  
**Instituto de Ciências Matemáticas e de Computação**  
**Departamento de Ciências de Computação**  
**Disciplina de Organização de Arquivos**  
**Profa. Dra. Cristina Dutra de Aguiar Ciferri**

**Trabalho Prático**

*Esse trabalho deve ser feito em grupo de no mínimo 3 alunos e no máximo 4 alunos. A solução deve ser proposta exclusivamente pelo grupo com base nos conhecimentos adquiridos ao longo das aulas. Consulte as notas de aula e o livro texto quando necessário.*

---

**Descrição**

Considere os seguintes dados a respeito de uma pessoa:

KEY (i.e., número que identifica a pessoa)  
LASTNAME (i.e., sobrenome da pessoa)  
FIRSTNAME (i.e., primeiro nome da pessoa)  
ADDRESS (i.e., endereço com logradouro, número e complemento)  
CITY (i.e., cidade)  
STATE (i.e., sigla do estado com 2 caracteres, tal como SP)  
ZIP (i.e., CEP, tal como 222222-222)  
PHONE (i.e., número do telefone com DDD, tal como (022)2222-2222)

Existem diversos métodos para representar os conceitos de campo e de registro em um arquivo de dados, conforme explicado em sala de aula e também constante nas páginas 96 a 107 do livro *File Structures (second edition)*, de Michael J. Folk e Bill Zoellick.

Considere os seguintes tipos de arquivos de dados que representam os conceitos de campo e de registro conforme abaixo especificado:

Turma A

- Tipo 1: campos de tamanho fixo em bytes + registros de tamanho fixo em bytes
- Tipo 2: indicador de tamanho para campos + indicador de tamanho para registros

Turma B

- Tipo 1: delimitadores entre campos + registros de tamanho fixo em bytes
- Tipo 2: delimitadores entre campos + delimitadores entre registros

Implemente um programa, para cada um dos tipos de arquivo de dados, que:

[1] Permita a gravação dos dados de várias pessoas até que uma condição de parada seja detectada.

[2] Permita a recuperação dos dados, de todas as pessoas, armazenados no arquivo de dados, mostrando os dados de forma organizada na saída padrão para permitir a distinção dos campos e registros. Isso é equivalente a um *browsing* nos registros do arquivo de dados. Ou seja, deve-se permitir a visualização dos registros, um por vez.

[3] Permita a recuperação dos dados de uma pessoa, a partir do número do registro fornecido como entrada (apenas para o Tipo 1). Note que esse número não é o valor do campo KEY. O primeiro registro corresponde ao registro de número zero, o segundo registro corresponde ao registro de número 1 e assim por diante.

**Turma A** [4] Permita a recuperação dos dados de uma pessoa com base no valor do campo KEY por meio da busca sequencial.

**Turma B** [4] Permita a recuperação dos dados de uma pessoa com base no valor do campo KEY por meio de um índice de chave primária.

**Turma A** [5] Permita a recuperação dos dados de uma pessoa com base no valor do campo FIRSTNAME por meio de um índice de chave secundária.

**Turma B** [5] Permita a recuperação dos dados de uma pessoa com base no valor do campo LASTNAME por meio de um índice de chave secundária.

[6] Permita a remoção lógica de registros.

[7] Permita a inserção de registros adicionais, com reaproveitamento do espaço de registros logicamente removidos. O reaproveitamento deve ser feito de forma dinâmica. Deve-se levar em consideração questões de desempenho na implementação do reaproveitamento de espaço. A escolha da estratégia de reaproveitamento de espaço para cada tipo de arquivo de dados deve ser devidamente justificada.

**Restrições** As seguintes restrições têm que ser garantidas no desenvolvimento do trabalho.

- Os arquivos de dados devem ser gravados em disco no modo binário. O modo texto não deve ser usado.
- Os dados a serem inseridos no arquivo e utilizados nas buscas devem refletir dados do mundo real. Dica: procure na internet por geradores de bancos de dados sintéticos (ou desenvolva o seu).
- O número que identifica uma pessoa (KEY) deve ser um campo de chave primária.
- Os demais campos devem representar campos de chave secundária.
- Todo código fonte deve ser documentado. A documentação inclui, dentre outros, a documentação de procedimentos, de funções, de variáveis, de partes do código fonte que realizam tarefas específicas. Ou seja, o código fonte deve ser documentado tanto em nível de rotinas quanto em nível de variáveis e blocos funcionais. Ademais, os integrantes do grupo devem constar como comentário no início do código (i.e., NUSP e nome de cada integrante do grupo). Não será atribuída nota ao aluno cujos dados não constarem no código fonte.
- A implementação deve ser realizada usando a linguagem de programação C e as funções da biblioteca <stdio.h> ou na linguagem de programação C++ e a biblioteca equivalente. A implementação não deve ser feita em qualquer outra linguagem de programação. Detalhes sobre o compilador a ser utilizado serão disponibilizados na página da disciplina.

### Material a ser Entregue

- **Programa.** Enviar por email para o aluno PAE da disciplina o arquivo compactado contendo o código fonte do programa, o código executável do programa, as bibliotecas necessárias para a execução do programa e a documentação externa em formato PDF. A mensagem deve ter como assunto: Trabalho Prático de Organização de Arquivos. Deve constar no corpo da mensagem o NUSP e nome de cada integrante do grupo. Não será atribuída nota ao aluno cujos dados não constarem no corpo da mensagem.
- **Impressão da Documentação Externa.** Essa documentação deve conter a descrição dos principais conceitos usados no trabalho prático, as decisões de projeto e as suas justificativas (ex.: estruturas de dados e algoritmos usados), assim como qualquer outra consideração adicional assumida no desenvolvimento do trabalho prático. A capa da documentação externa deve conter as seguintes informações: o nome da instituição, o nome do curso, o nome da disciplina, o nome do professor responsável, o nome do trabalho prático, o nome dos participantes e os respectivos números, e a data de entrega do trabalho prático. A documentação externa também deve conter um índice que lista os nomes das seções que compõem o trabalho prático e as suas respectivas páginas de início.
- **Impressão do Código Fonte.** A documentação externa também deve conter a impressão do código fonte devidamente documentado e numerado linha a linha.
- **Impressão de Baterias de Testes e Cópias da Interface.** A documentação externa também deve conter a impressão de uma ou mais baterias de teste, as quais devem gerar informações que permitam acompanhar a execução do programa. Também devem ser incluídas cópias de telas da interface.

**Análise do Trabalho** Na correção do trabalho, serão ponderados os seguintes aspectos.

- Qualidade da documentação entregue.
- Corretude da execução do programa.
- Eficiência da implementação.
- Qualidade da interface.

Também serão consideradas as seguintes restrições.

- A não execução de um programa devido a erros de compilação implica que a nota final do trabalho prático será igual a zero (0).
- A ausência da documentação interna implica que haverá uma diminuição expressiva na nota do trabalho prático.
- A ausência da documentação externa implica que haverá uma diminuição expressiva na nota do trabalho prático.
- A inserção de palavras ofensivas nos arquivos e em qualquer outro material entregue implica que a nota final do trabalho prático será igual a zero (0).
- Em caso de cola, as notas dos trabalhos envolvidos será igual a zero (0).
- Devem ser exibidos avisos ou mensagens de erro quando apropriado.

**Divisão do Trabalho Prático** De acordo com o critério de avaliação da disciplina, o trabalho prático é dividido em duas partes ( $PR_1$  e  $PR_2$ ). Essa divisão é feita da seguinte forma:

$PR_1$ : itens [1] a [5], para os tipos de arquivos de dados 1 e 2.

$PR_2$ : itens [6] a [7], para os tipos de arquivos de dados 1 e 2.

**Datas de Entrega do Trabalho Prático** Nas datas especificadas na página da disciplina.

**Bom Trabalho !**