**PRÁTICA LABORATORIAL 04**

|  |
| --- |
| **Objetivos:**   * Arrays de Strings |

EXERCÍCIOS – GRUPO I

**ENUNCIADOS DOS EXERCÍCIOS**

**EXERCÍCIO 1: LISTA DE COMPRAS**

**Objetivo:** Criar um programa que gere uma lista de compras simples.

Enunciado: Escreve um programa em C que:

1. Permita ao utilizador inserir até 5 itens para uma lista de compras
2. Mostre a lista completa numerada (1. item1, 2. item2, etc.)
3. Conte e mostre o total de itens na lista
4. Encontre e mostre o item com o nome mais longo
5. Pergunte ao utilizador por um item específico e diga se está ou não na lista

**EXERCÍCIO 2: REGISTO DE ESTUDANTES**

**Objetivo:** Criar um registo simples de nomes de estudantes.

Enunciado: Escreve um programa em C que:

1. Permita inserir os nomes de até 8 estudantes numa turma
2. Mostre todos os nomes numa lista organizada
3. Conte quantos nomes começam com cada uma das letras: A, B, C
4. Encontre o nome mais curto e o nome mais longo
5. Calcule e mostre o número médio de letras por nome
6. Permita procurar um estudante pelo nome completo

**EXERCÍCIO 3: CATÁLOGO DE FILMES**

**Objetivo:** Criar um sistema de gestão de filmes com múltiplas funcionalidades.

Enunciado: Escreve um programa em C que implemente um catálogo de filmes:

1. Permita adicionar até 10 filmes (só o título)
2. Apresente um menu com as seguintes opções:
   * Adicionar filme
   * Listar todos os filmes
   * Procurar filme por nome completo
   * Procurar filmes que contenham uma palavra
   * Mostrar estatísticas
   * Sair
3. Na opção "Mostrar estatísticas", apresente:
   * Total de filmes no catálogo
   * Filme com título mais longo e mais curto
   * Número médio de caracteres por título
   * Quantos filmes contêm a palavra "de" no título
4. As pesquisas devem ser case-insensitive (não distinguir maiúsculas/minúsculas)
5. O programa deve continuar a funcionar até o utilizador escolher "Sair"

**EXERCÍCIO 4: ORGANIZADOR DE TAREFAS**

**Objetivo:** Criar um gestor de tarefas com prioridades.

Enunciado: Escreve um programa em C que gerencie uma lista de tarefas:

1. Permita adicionar até 15 tarefas, cada uma com:
   * Descrição da tarefa
   * Prioridade ("Alta", "Media", "Baixa")
2. Implemente um menu com opções:
   * Adicionar nova tarefa
   * Listar todas as tarefas
   * Listar apenas tarefas de prioridade "Alta"
   * Ordenar tarefas alfabeticamente
   * Contar tarefas por prioridade
   * Procurar tarefa por palavra-chave
   * Sair
3. Ao listar tarefas, mostre o formato: "Prioridade: Descrição"
4. Na opção "Contar tarefas por prioridade", mostre quantas tarefas existem de cada tipo
5. A pesquisa por palavra-chave deve mostrar todas as tarefas que contenham essa palavra
6. Valide que a prioridade inserida é uma das três opções válidas

**EXERCÍCIO 5: AGENDA TELEFÓNICA**

**Objetivo:** Criar uma agenda telefónica simples com funcionalidades essenciais.

Enunciado: Escreve um programa em C que implemente uma agenda telefónica:

1. Permita gerir até 10 contactos, cada um com:
   * Nome da pessoa
   * Número de telefone
2. Implemente um menu com as seguintes opções:
   * Adicionar contacto
   * Listar todos os contactos
   * Procurar contacto por nome
   * Ordenar contactos por nome (A-Z)
   * Mostrar estatísticas
   * Sair
3. Na opção "Mostrar estatísticas", apresente:
   * Total de contactos guardados
   * Nome mais longo e mais curto
   * Quantos contactos começam com a letra "A"
   * Média de caracteres por nome
4. A pesquisa por nome deve mostrar o formato: "Nome: Telefone"
5. Implemente validações básicas:
   * Não permitir nomes vazios
   * Não exceder o limite de 10 contactos
   * Não permitir adicionar o mesmo nome duas vezes
6. Ao ordenar os contactos, certifique-se que os números de telefone acompanham os nomes correspondentes