Universidade do Estado de Santa Catarina (UDESC)
Curso: Tecnologia em Análise e Desenvolvimento de Sistemas
Disciplina: Estrutura de dados (EDA0001)
Prof. Rui Jorge Tramontin Jr.

Trabalho 3: Agenda de Eventos usando Lista Encadeada

O objetivo do trabalho é desenvolver um programa que gerencie um cadastro simples de uma **agenda de eventos**. Quando inicializado, o programa deve carregar os dados de um **arquivo de texto** em uma **lista encadeada de estruturas do tipo evento**, cujo modelo é apresentado a seguir.

1. **Tipo *Evento*** (*estrutura* contendo os campos a seguir):
   - data (**Tipo *Data***);
   - horário de início (**Tipo *Hora***);
   - horario de fim (**Tipo *Hora***);
   - descrição (*string* com 50 caracteres);
   - local (*string* com 50 caracteres);

2. **Tipo *Data*** (*estrutura* contendo *dia*, *mes*, *ano*);

3. **Tipo *Hora*** (*estrutura* contendo *hora*, *minuto*).

Caso o *arquivo não exista*, o programa deve criar uma **lista vazia**. O programa deve oferecer um **menu de texto** com as seguintes opções:

1. **Cadastrar** (somente um) novo evento na lista;
    - A inserção deve ser feita **em ordem**; implemente uma ***função de comparação*** (primeiro por data e, caso tenha a mesma data, por horário de início);
    - Além disso, é preciso garantir que o novo evento não sobreponha o intervalo (horários de início e de fim) de outro evento já cadastrado (é preciso fazer uma busca na lista → analise exemplos para definir a lógica da função de comparação);

2. **Mostrar todos os eventos da agenda**. Utilize a função para mostrar a lista (usando uma função para mostrar um evento);

3. Dada uma data, **mostrar todos os eventos dessa data**. *Dica: utilize a função da questão 3, da lista de exercícios 5 (com uma função de comparação por data)*;

4. Dada uma descrição, **mostrar todos os eventos que tenham essa descrição**;
    - *Dica: novamente, utilize a função da questão 3, da lista de exercícios 5 (com uma função de comparação pela descrição);*
    - Para simplificar, a busca pode ser pela *string* exata, ou seja, a função de comparação pode ser feita com *strcmp()*;

5. **Remover evento**; aqui devem ser dadas 2 opções ao usuário:
    a. Dada uma data, remover todos os eventos dessa data (ou seja, pode remover mais de um evento);
    b. Data uma data e hora inicial, remover o evento (ou seja, remove um único evento);

6. **Sair do programa**: salva a lista no arquivo de texto e desaloca a lista.

Requisitos do programa

- Validação da entrada (data e hora com valores válidos);
- Cuidado com a interface de usuário: coloque mensagens apropriadas para situações excepcionais, tais como erro na validação da entrada ou que o evento já existe (opção 1), lista de eventos vazia (opções 2, 3, 4), ou que o evento não foi encontrado (opção 5);
- Uso correto da biblioteca de listas, com funções de comparação apropriadas;
- Boas práticas no uso do encapsulamento dos tipos abstratos de dados;
- Não se esqueça de desalocar a lista ao final do programa!

Critérios de avaliação

- Execução correta e alinhamento com o que foi solicitado neste enunciado.

Informações importantes:

- **Equipe:** 1 ou 2 alunos.
- **Entrega via Moodle.**