



Trabalho prático 1 – Estruturas de Dados I

Data de entrega: 10/05/21

OBSERVAÇÕES:

1. Plágio não será tolerado. Trabalhos iguais serão penalizados com **nota zero**.
2. O trabalho poderá ser feito **em duplas (no máximo)**.
3. Você pode acrescentar funções que julgar necessárias para completar as tarefas.
4. Você deve criar as estruturas **exatamente** como especificado nesse trabalho e criar as funções **com os mesmos protótipos dados**.
5. Não deixe de indentar e documentar seu código para facilitar o entendimento do trabalho.
6. Para entrega do trabalho, o aluno deverá compactar a pasta contendo o projeto e todos os seus arquivos e **dar o nome da dupla** ao arquivo compactado. Enviar o arquivo compactado via Classroom.

Você deve implementar um sistema para manter um cadastro de times. Posteriormente, será necessário sortear times para se enfrentar em jogos de uma competição. A sua primeira tarefa será implementar algumas funcionalidades isoladas do programa que permitam cadastrar e manipular os dados de times.

No seu programa você deve implementar os **TADs Time e Lista**. A lista implementada deve ser uma **lista duplamente encadeada com descritor de times**. Assim, o seu projeto deve conter pelo menos os seguintes arquivos: Time.h, Time.c, Lista.h, Lista.c, main.c e makefile.

Um time possui um **identificador único** (deve ser incrementado automaticamente), um **nome** e sua **origem**. Na implementação do TAD correspondente, você deve incluir **pelo menos** as funções para criação, edição e impressão de **um** time.

Ex: protótipo da função criar time: `Time* criarTime();` // Essa função deve alocar espaço para um time e pedir ao usuário as informações para preenchê-lo (exceto identificador que deve ser incluído automaticamente).

O seu programa principal deve manter a lista de times. O campo info do nó da lista deve ser **uma estrutura** do tipo Time e as funções devem receber essa estrutura.

Ex: protótipo da função insere na lista duplamente encadeada: `void insereOrdenado(Descritor* d, Time v);`

A lista de times deve ser mantida **ordenada pelo nome** do time.

As alterações das listas estudadas em sala de aula para atender a especificação do trabalho são de responsabilidade dos alunos.

Implemente sua função principal (*main*) no arquivo main.c. Ela deve criar a lista de times e, em seguida, imprimir na tela um menu com as seguintes opções:

-----MENU-----

- 1 - Cadastrar time
- 2 - Editar time
- 3 - Remover time
- 4 - Imprimir times
- 5 - Sortear grupos
- 6 - Imprimir grupos
- 7 - Sair

Escolha uma opção:

Implemente todas as funcionalidades do menu, levando em conta as seguintes considerações:

- Para cadastrar novo time (opção 1), deve-se criar a estrutura correspondente e inserir na lista.
- A busca de um time deve ser feita pelo identificador único (tanto na opção 2, quanto na opção 3).
- Na opção 3, deve ser impresso um time por linha.
Ex: 1 – Flamengo – Rio de Janeiro
2 – Vasco – Rio de Janeiro
- Na opção 5, devem ser gerados 2 grupos contendo cada grupo a metade dos times existentes na lista. Os grupos serão formados a partir de um sorteio tendo como base os times previamente cadastrados na lista. Cada grupo deve ser uma lista cujos nós possuem apenas o identificador do time.