Exercício Avaliativo

ID: E1-2025

Problema

Um aplicativo agregador de músicas pediu sua ajuda para desenvolver um sistema de playlist, capaz de gerenciar as músicas inclusas nela.
Desenvolva dois TADs envolvendo um tipo Musica e um tipo Playlist, bem como uma main para executar suas funcionalidades.

Uma Musica possui um nome, artista, album e gênero musical. Uma Playlist possui um nome, um vetor de ponteiros de Musicas (Musicas**) e uma quantidade de músicas salvas.

TAD Musica: Possui funções de criação de música, comparação de duas músicas, comparações de atributos (ex: comparaArtistaMus(Musica*, char*)) e diferentes funções para imprimir músicas – diferentes formas de imprimir uma músicas. Possui também funções de callback de comparação de atributo (para apontar pra uma função de comparação de atributo ex: comparaArtistaMus) e funções de callback para apontar para funções de impressão – já que existe várias (4). O TAD Musica possui as seguintes funções:

- **Musica* criaMusica()**: aloca e inicializa um ponteiro do tipo Musica. Os ponteiros internos do tipo musica (nome, artista, etc.) devem receber NULL.
- **Musica* leMusica(Musica *music)**: recebe um ponteiro Musica* já inicializado. Lê da entrada as informações que vão ser atribuidas à música, na ordem: nome, artista, album e genero. Aqui ocorre a alocação de fato desses ponteiros. Retorna o ponteiro Musica com as informações inicializadas.
- **int comparaMusicas(Musica *music_1, Musica *music_2)**: verifica se duas músicas são iguais, usando como parâmetro o nome E o artista das músicas em questão. Retorna 1 se verdadeiro, 0 caso contrario.
- **int comparaNomeMus(Musica *music, char *name)**: verifica se uma música possui nome igual ao passado como parâmetro. Retorna 1 se verdadeiro, 0 caso contrario.
- **int comparaArtistaMus(Musica *music, char *artist)**: verifica se uma música possui artista igual ao passado como parâmetro. Retorna 1 se verdadeiro, 0 caso contrario.
- **int comparaAlbumMus(Musica *music, char *album)**: verifica se uma música possui album igual ao passado como parâmetro. Retorna 1 se verdadeiro, 0 caso contrario.
- **int comparaGeneroMus(Musica *music, char *genre)**: verifica se uma música possui genero igual ao passado como parâmetro. Retorna 1 se verdadeiro, 0 caso contrario.

- **void imprimeMusica(Musica *music)**: imprime as informações da música na seguinte formatação: nome - artista ..... album, com uma quebra de linha no final.
- **void imprimeMusica_Album(Musica *music)**: imprime o album da música.
- **void imprimeMusica_Artista(Musica *music)**: imprime o artista da música.
- **void imprimeMusica_Genero(Musica *music)**: imprime o genero da música.
- **void apagaMusica(Musica *music)**: função free que desaloca um tipo Musica.

TAD Playlist: Possui funções de criação de playlist, adição de música em playlist, remoção de música na playlist e impressão de listas de música. A principal finalidade do TAD é de trabalhar a utilização dos ponteiros para funções. isso acontece na main e no tad playlist através da utilização da função impressFunction.
O TAD Playlist possui as seguintes funções:

- **Playlist *criaPlaylist(char *name)**: cria uma playlist vazia com todas as posicoes do array de musicas como NULL.
- **void adicionaMusica(Playlist *playlist)**: adiciona uma musica em uma playlist.
- **int removeMusica(Playlist *playlist, char *music, compara comparador, tipoImpressaoMusica impMus)**: remove uma musica de uma playlist. O parametro "comparador"é um ponteiro para função(callback) de uma função de comparação de música. O parâmetro "impMus"dessa função é um ponteiro (callback) para uma função de impressão. Deve-se imprimir as informações da música removida com essa função.
- **void impressFunction(Playlist *playlist, char *key, compara comparador, tipoImpressaoMusica impMus)**: Função genérica que realiza impressões filtradas na playlist (todas as impressoes sao feitas por essa funcao). O parametro key é a chave de busca para filtragem (nome do artista, por exemplo). Dica: ponteiros para função também podem ser NULL.
- **void apagaPlaylist(Playlist *playlist)**: função free que desaloca um tipo Playlist.

Main: O funcionamento do programa será dado da seguinte forma: na primeira linha de entrada do terminal será escrito o nome da playlist e na linha seguinte a quantidade de ações que serão feitas sobre essa playlist (inseri música, remover, imprimir, etc...). Depois disso, para cada ação deve-se inserir um número específico da ação.

1 - Inserir música cujos dados devem ser informados na linha seguinte no formato da leitura de música;
2 - Remover música cujo o nome deve ser informado na linha seguinte;
3 - Imprimir playlist;
4 - imprimir uma lista das músicas de um artista que estão nessa playlist (o nome do artista deve ser inserido na linha seguinte);
5 - imprimir todas as músicas de um mesmo álbum que estão nessa playlist (o nome do álbum deve ser inserido na linha seguinte);
6 - Imprimir todas as músicas de um mesmo gênero que estão nessa playlist (o gênero deve ser inserido na linha seguinte).

Casos de Teste

Verifique os casos de entrada e saida para melhor compreender a formatação.