UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO – CAMPUS PAU DOS FERROS
BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO
DISCIPLINA: LABORATÓRIO DE ALGORITMOS E ESTRUTURAS DE DADOS I

EXERCÍCIOS - TIPOS ABSTRATOS DE DADOS

Questão 1) Crie um TAD chamado ContaBancaria que possui os seguintes campos: titular, número e saldo. Como operações, considere as seguintes:

- Cria conta: aloca dinamicamente uma estrutura do tipo ContaBancaria e retorna seu endereço. Os campos titular, número e saldo devem ser fornecidos como parâmetros;
- Deposita: recebe, como parâmetros, o endereço de uma estrutura do tipo ContaBancaria e um valor, atualizando o saldo;
- Saca: recebe, como parâmetros, o endereço de uma estrutura do tipo ContaBancaria e um valor, atualizando o saldo. Verificar se o saldo é suficiente para realizar o saque;
- Transfere: recebe, como parâmetros, os endereços das estruturas do tipo ContaBancaria e um valor, atualizando os saldos. Verificar se o saldo é suficiente para realizar a transferência;
- Saldo: recebe o endereço de uma estrutura do tipo ContaBancaria e retorna seu saldo;
- Exclui conta: libera o espaço alocado dinamicamente para a estrutura.

Faça o que se pede nos itens a seguir:

a) Crie um arquivo (*contabancaria.h*) com a interface do TAD.
b) Crie um arquivo (*contabancaria.c*) com a implementação do TAD.
c) Crie um programa que utiliza o TAD ContaBancaria.

Questão 2) Você foi contratado por uma instituição educacional para desenvolver um sistema de gerenciamento de alunos e disciplinas. Para isso, você precisa criar um conjunto de Tipos Abstratos de Dados (TADs) que permitam a manipulação eficiente dos dados dos alunos e das disciplinas oferecidas pela instituição.

TAD Disciplina:

Crie um TAD chamado "Disciplina" com os seguintes campos e operações:

Campos:

- `char nome[100]`: armazena o nome da disciplina.
- `int codigo`: armazena o código identificador único da disciplina.

Operações:

- `Disciplina* cria_disciplina(char nome[], int codigo)`: aloca dinamicamente uma estrutura do tipo Disciplina e retorna o seu endereço. Os campos `nome` e `codigo` devem ser fornecidos como parâmetros.
- `void exclui_disciplina(Disciplina* disciplina)`: libera o espaço alocado dinamicamente para a estrutura da disciplina.

TAD Aluno:

Crie um TAD chamado "Aluno" com os seguintes campos e operações:

Campos:

- `char nome[100]`: armazena o nome do aluno.
- `int matricula`: armazena o número de matrícula único do aluno.
- `Disciplina* disciplinas[10]`: um array de ponteiros para Disciplinas, que armazena as disciplinas em que o aluno está matriculado.
- `int num_disciplinas`: armazena o número atual de disciplinas em que o aluno está matriculado.

Operações:

- `Aluno* cria_aluno(char nome[], int matricula)`: aloca dinamicamente uma estrutura do tipo Aluno e retorna o seu endereço. Os campos `nome` e `matricula` devem ser fornecidos como parâmetros.
- `void matricula_disciplina(Aluno* aluno, Disciplina* disciplina)`: matricula o aluno em uma disciplina, adicionando o ponteiro da disciplina ao array de disciplinas do aluno.
- `void exclui_aluno(Aluno* aluno)`: libera o espaço alocado dinamicamente para a estrutura do aluno.

Programa de Utilização:

Crie um programa que utilize os TADs Disciplina e Aluno. Seu programa deve permitir a criação de disciplinas e alunos, a matrícula de alunos em disciplinas, e a exibição das informações dos alunos e das disciplinas em que estão matriculados.

Lembre-se de criar os arquivos `disciplina.h`, `disciplina.c`, `aluno.h`, `aluno.c` e um arquivo `main.c` para a execução do programa.

Observações:

- Certifique-se de verificar os limites de matrícula e disciplinas conforme necessário.
- Implemente verificações adequadas para evitar duplicação de disciplinas no array de disciplinas do aluno.
- Garanta que todos os recursos alocados dinamicamente sejam devidamente liberados para evitar vazamentos de memória.
- Você pode usar as funções e estruturas da linguagem C para implementar esses TADs e o programa de exemplo.

*Fonte:
CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas. **Introdução a Estruturas de Dados: com técnicas de programação em C**. Elsevier, 2016