



UNIVERSIDADE FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS
DS-130 - ESTRUTURAS DE DADOS I – 3º. PERÍODO
TRABALHO PRÁTICO I

Postagem do Trabalho no Moodle – 04/04/2022 (segunda)

Defesas do Trabalho: 05 e 12/04/2022

PROBLEMA 1 – CALCULADORA DE MATRIZ ESPARSA (50 pontos)

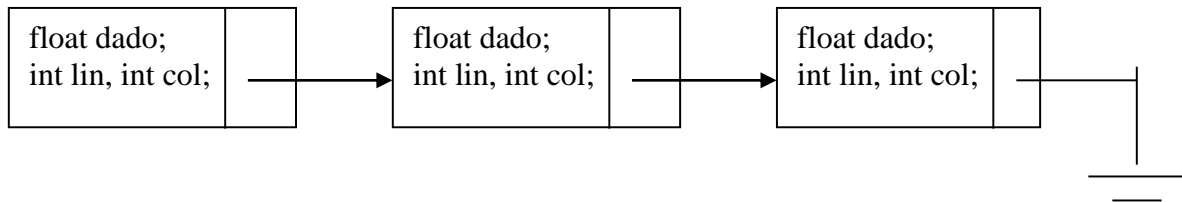
Criar um programa em C que manipule matrizes esparsas, utilizando lista linear encadeada.

OBS: Uma matriz é considerada esparsa quando a maioria dos seus dados são zeros.

Exemplo de Matriz Esparsa

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3.5
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	5	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	0

Em caso da matriz ser muito grande não é vantajoso, em termos de memória, armazenar todos os dados, até porque a maioria são zeros. Então o objetivo do trabalho é criar uma lista encadeada e armazenar somente os dados diferentes de zero, bem como a linha e a coluna onde esses dados estão localizados.



Mas, no entanto, o programa deverá imprimir na saída a matriz inteira, tanto os dados diferentes de zero como os zeros.

O programa deverá ser formado pela seguinte struct:

```

typedef struct nodo
{
    float dado;
    int lin, col;
    struct nodo *prox;
}Matriz_Esparsa;
  
```

E pelas seguintes funções:

- Uma função que faz a alocação de memória para cada nodo criado para uma lista encadeada;
- Uma função que insere na lista encadeada um nodo alocado;
- Uma função que busca os dados em uma lista encadeada;
- Uma função que libera da memória uma lista encadeada;
- Uma função que lê os dados da matriz, via teclado, e insere na lista encadeada somente os dados diferentes de zero;
- Uma função que soma duas matrizes;
- Uma função que subtrai duas matrizes;
- Uma função que multiplica duas matrizes;
- Uma função que gera a matriz transposta;
- Uma função que imprime todos os dados da matriz, inclusive os zeros;
- Uma função que imprime os elementos da diagonal principal, inclusive os zeros caso existam.

Observações:

- O usuário pode entrar com quantas matrizes desejar.
- O usuário deve informar a dimensão de cada matriz.
- O usuário poderá fazer quantas operações desejar com as matrizes.
- Exemplo: **MA * MB + MC - MD**
- Considerar as regras de operações com matrizes para implementar a soma, subtração, multiplicação, transposta e diagonal principal de matrizes.
- As listas encadeadas (que representam as matrizes) devem ser passadas como parâmetros das funções.

Avaliação:

- **(2.0 ponto)** Implementação de todas as funções
 - **(1.5 ponto)** programa rodando
 - **(1.5 pontos)** defesa individual
-

PROBLEMA 2 – ROTAS DE ÔNIBUS (50 pontos)

Objetivo: Implementar uma Lista de Rotas de Ônibus utilizando Listas Duplamente Encadeadas.

- Este programa deverá gerenciar um conjunto de listas de rotas/linhas de uma companhia de ônibus intermunicipal do Estado do Paraná;
- Todas as rotas iniciam em Curitiba;
- Cada rota possui um nome que a identifica de forma única, dado pelo destino. Ex: "Campo Largo";
- Cada rota possui a lista de todas as cidades por onde o ônibus passa, para ir de Curitiba ao destino;
- A rota de ônibus, ao retornar, passa pelos mesmos lugares. Portanto, pode ser representada por uma lista duplamente encadeada;
- O usuário deve poder escolher uma rota e poder navegar por ela, indo de cidade em cidade e voltando para a cidade anterior, usando para tanto as teclas de seta à esquerda e seta à direita ou 1- Ir para próxima cidade; 2 – Voltar para a cidade anterior;
- Cada nodo representando uma cidade possui, além do nome, um texto descrevendo alguma característica desta cidade. Ex.: "Lá encontram-se as melhores porcelanas". Este texto é mostrado quando se visita um nodo.
- Implemente o programa de rotas como uma lista de listas;
- A lista que contém todas as rotas pode ser uma lista encadeada ou uma lista estática em vetor em vetor;

- Cada lista que representa uma rota deve ser uma lista duplamente encadeada;
- Aplicar o conceito de nó descritor na lista duplamente encadeada.
- Menu de Entrada:
 1. Cadastrar Rota
 2. Excluir Rota
 3. Visitar Rota
 4. Sair

Avaliação:

- **(2.0 ponto)** Implementação de todas as funcionalidades
- **(1.5 ponto)** programa rodando
- **(1.5 pontos)** defesa individual

ATENÇÃO:

1. Equipes de no máximo dois alunos; em caso de cópias de trabalhos todas as equipes envolvidas receberão nota zero.
2. Questões sobre o trabalho podem ser solicitadas na 1ª prova teórica.