

Data de entrega: 23/11/2011

Leia atentamente todas as instruções abaixo:

- 1) Um (1,0) ponto será descontado na nota para cada dia de atraso na entrega.
- 2) O trabalho deve ser desenvolvido em grupos de 2 pessoas.
- 3) Enviar o trabalho para o email do docente:
lantiq@icmc.usp.br ou lantiq@gmail.com (assunto da mensagem: "[ALG-I] Entrega TP2")
Uma mensagem de confirmação de recebimento será enviada pelo professor ao(s) aluno(s).
- 4) Não anexar no email arquivos executáveis. Inclusive, verifique todas as sub-pastas de arquivos do tipo zip, rar, etc.
- 5) Não deixe para entregar o trabalho na última hora. A responsabilidade de entrega é do aluno, e não do servidor de email. A entrega por email facilita o processo, mas não garante 100% de confiabilidade. Se, por algum motivo, o servidor de emails estiver com problemas e você não receber a mensagem de confirmação de recebimento, grave o trabalho em um CD e entregue-o na sala do docente (3-160). Nesse caso, identifique o CD com os nomes dos alunos, o conteúdo e a data. Deixe o CD embaixo da porta, caso o docente não se encontre na sala. Ou seja, faça com que o trabalho chegue às mãos do docente.

Especificação do Trabalho

Escreva um programa que implemente o **TAD Matriz Esparsa**, o qual deve disponibilizar as seguintes operações:

1. Definir(M , arquivo): lê de um arquivo texto os elementos não nulos de uma matriz e os armazena em M .
2. Somar(M , $M1$, $M2$): retorna em M a matriz resultante da soma de duas matrizes $M1$ e $M2$. As matrizes $M1$ e $M2$ devem ter sido anteriormente lidas de arquivos texto utilizando a função DefinirMatriz.
3. SomarConstante(M , c , i): soma a constante c a todos os elementos da linha i da matriz M .
4. Imprimir(M): imprime completamente a matriz M , linha a linha.
5. Inserir(M , i , j , x): insere na matriz M o valor x na linha i e coluna j . Se já existir um valor na posição, adicionar o valor x a ele. Se o valor adicionado gerar um valor final nulo, o respectivo nó deve ser removido da matriz.
6. Buscar(M , i , j): retorna o elemento da linha i e da coluna j da matriz M , mesmo que seja um valor nulo.
7. Finalizar(M): elimina a matriz M , removendo (desalocando) todos os elementos nela contidos.

Seu programa deve usar a representação de matrizes esparsas por listas cruzadas vista em aula:

```
typedef struct no { //Estrutura para cada elemento não nulo
    int lin, col, val; //Conteúdo (val) da linha lin e coluna col
    struct no *proxlin; //Próximo elemento na mesma linha
    struct no *proxcol; //Próximo elemento na mesma coluna
} no;

typedef struct {
    no **L, **C; //Vetores de ponteiros para o início das listas
                // de linhas e colunas (esses vetores devem
                // ser alocados dinamicamente)
    int nlin, ncol; //Número de linhas e de colunas na matriz
} MatrizEsparsa;
```

O programa principal deve apresentar ao usuário um menu de opções, incluindo todas as operações indicadas anteriormente, além da opção de sair do programa. Portanto, o usuário deve ser capaz de realizar as seguintes operações, quantas vezes desejar:

- (i) Definir duas matrizes de mesma dimensão, lendo os respectivos elementos de dois arquivos. Os nomes dos arquivos devem ser fornecidos pelo usuário.
- (ii) Somar essas duas matrizes e armazenar em uma terceira.
- (iii) Somar uma constante a uma linha de qualquer uma dessas três matrizes.
- (iv) Imprimir qualquer uma das três matrizes.
- (v) Inserir um elemento em qualquer uma das três matrizes.
- (vi) Buscar um elemento em uma posição específica de qualquer uma das três matrizes.

Desaloque memória sempre que não for mais utilizar. Além disso, faça checagem de erros quando necessário.

O arquivo texto de uma matriz deve estar em um formato específico, exemplificado abaixo:

3 3
0 2 12
1 1 5.9
2 1 -3

Onde a primeira linha indica a dimensão da matriz (no exemplo acima, a matriz tem 3 linhas e 3 colunas). As linhas seguintes apresentam três números: i, j e k, os quais indicam que o valor k deve ser inserido na posição (i,j) da matriz. O exemplo acima representa a seguinte matriz:

$$\begin{bmatrix} 0 & 0 & 12 \\ 0 & 5.9 & 0 \\ 0 & -3 & 0 \end{bmatrix}$$

Você deverá entregar:

- 1) O código-fonte, com comentários explicando as principais funcionalidades do seu programa. Também preste atenção à correta indentação de seu código.
- 2) A documentação do seu trabalho, incluindo:
 - a. Passo-a-passo para compilação do projeto.
 - b. Explicação detalhada dos algoritmos utilizados no TAD. Caso precise, faça figuras esquemáticas.