



Instituto de Computação
Universidade Estadual de Campinas

MC202 - Estruturas de Dados



Laboratório 1

Alocação dinâmica de memória

Data de publicação: Sexta feira, 4 de março de 2016

Prazo máximo de submissão: Sexta feira, 11 de março de 2016 às 23h59m

Professor: Neucimar J. Leite <neucimar@ic.unicamp.br>

Monitores:

- Juan Hernández (PED) <juan.albarracin@students.ic.unicamp.br>
- Leonardo Yvens (PAD) <leoyvens@gmail.com>

Grupo do curso: https://groups.google.com/d/forum/mc202bc_2016s1

Enunciado

Implemente em C as operações matriciais de soma, produto, transposta e ponto de sela como descritas abaixo:

Operador	Entrada	Especificação	Resultado esperado
Soma	Matrizes A e B	Implementar uma função que calcule a soma termo a termo das matrizes A e B como especificado na entrada correspondente da Wikipédia .	Matriz resultante da soma ou ponteiro nulo.
Multiplicação	Matrizes A e B	Implementar uma função que calcule o produto $A \times B$ como especificado na entrada correspondente da Wikipédia .	Matriz resultante da multiplicação ou ponteiro nulo.
Transposta	Matriz A	Implementar uma função que calcule a matriz	Matriz

		transposta de A como especificado na entrada correspondente da Wikipédia .	transposta de A.
Ponto de sela	Matriz A	Um elemento A_{ij} de uma matriz é dito ser um ponto de sela da matriz A se for, ao mesmo tempo, o menor elemento da linha i e o maior elemento da coluna j . Deve-se implementar uma função que encontre as coordenadas do primeiro ponto de sela encontrado, percorrendo a matriz desde o canto superior esquerdo primeiramente pelas linhas (A_{00} , A_{01} , A_{02} , ..., A_{10} , A_{11} , ...).	Linha e coluna do primeiro ponto de sela encontrado, se ele existir.

Os cabeçalhos dos métodos referentes às operações a serem implementadas encontram-se no arquivo **oper.h**. Não é permitido modificar sua estrutura. Segue uma breve descrição dos mesmos:

```
int **soma(int **A, int l_A, int c_A, int **B, int l_B, int c_B);
```

A função soma retorna um ponteiro duplo à matriz resultante. O fato de receber dois ponteiros duplos às matrizes de entrada indica que todas as coleções de dados (arrays e matrizes) **devem ser alocadas dinamicamente** e portanto tratadas como ponteiros. Adicionalmente, a função deve receber as dimensões de cada matriz. No cabeçalho, por exemplo, os inteiros l_A e c_A correspondem respectivamente ao número de linhas e colunas da matriz A.

Deve-se considerar também o caso em que não seja possível fazer a soma, devido a inconsistência nas dimensões das duas matrizes. Caso aconteça, a função deve retornar um ponteiro nulo.

```
int **multiplicacao(int **A, int l_A, int c_A, int **B, int l_B, int c_B);
```

De maneira similar à soma, a função multiplicação recebe e retorna ponteiros às matrizes. Igualmente, deve-se considerar o caso de inconsistência nas dimensões das matrizes para fazer a multiplicação, retornando um ponteiro nulo.

```
void ponto_sela(int **A, int l_A, int c_A, int *linha, int *coluna);
```

Dado que o ponto de sela consiste em dois valores inteiros (coordenadas do valor encontrado), o método que deve ser implementado não deve retornar nenhum valor, mas atualizar o valor de duas variáveis que foram passadas por referência, correspondentes à linha e coluna do ponto de sela.

```
int **transposta(int ** A, int l_A, int c_A);
```

O valor retornado sempre deve ser o ponteiro duplo à matriz A transposta.

Especificação de entrada e saída

Para ser avaliado, o programa desenvolvido deve fazer leitura dos dados de entrada e a posterior escrita dos resultados. Segue a estrutura da entrada ao programa:

- A primeira linha contém um caractere que identifica a operação a ser executada:

a	-	Soma
m	-	Multiplicação
s	-	Ponto de sela
t	-	Transposta

- A segunda linha contém dois números inteiros, separados por espaço:

m_A n_A

Os quais indicam as dimensões da primeira matriz.

- As seguintes m_A linhas contêm n_A números inteiros separados por espaço, representando cada linha da matriz.
- Se a primeira linha contiver os caracteres 's' ou 't', a entrada terá finalizado. Caso contrário, a seguinte linha terá os inteiros m_B n_B correspondentes às dimensões da segunda matriz.
- As seguintes m_B linhas contêm n_B números inteiros separados por espaço, representando cada linha da segunda matriz.

Segue a estrutura da saída:

- Se a primeira linha da entrada contiver os caracteres 'a', 'm' ou 't', a saída deverá consistir em m_R linhas contendo n_R inteiros separados por espaço, onde m_R e n_R são as dimensões da matriz resultante.
- Caso contrário, para o caractere 's', a saída deverá consistir em uma linha com dois inteiros separados por espaço, representando nessa ordem, a linha e a coluna do ponto de sela encontrado.
- O programador deve formatar adequadamente as saídas das funções sem resposta válida. Se a função de soma ou multiplicação retornarem um ponteiro nulo, a saída

do programa deve ser uma linha contendo o texto “NE”. Igualmente, se não existir um ponto de sela na matriz, a saída do programa deve ser “NE”.

Seguem alguns exemplos de entradas com suas respectivas saídas.

Operação	Entrada	Saída
Soma	a 3 4 1 2 3 4 5 6 7 8 9 10 11 12 3 4 2 4 6 8 10 12 14 16 18 20 22 24	3 6 9 12 15 18 21 24 27 30 33 36
Multiplicação	m 3 2 4 2 3 4 5 6 4 3 6 8 8 3 1 6 7 5 2 8 8 0	NE
Ponto de sela	s 5 5 4 6 2 9 1 8 9 4 9 9 3 4 1 6 5 2 3 3 7 2 1 5 3 9 9	1 2
Transposta	t 2 3 1 2 3 3 2 1	1 3 2 2 3 1

Estrutura da submissão

O código fonte deve ser submetido no sistema [SuSy](#) para ser executado e testado. O sistema receberá três arquivos:

Nome	Função
main.c	Programa principal que lê os dados de entrada, faz o chamados às funções e escreve a saída.
oper.h	Arquivo com os cabeçalhos descritos acima. Esse arquivo será fornecido inicialmente e, embora não seja permitido modificar a estrutura dos cabeçalhos presentes, permite-se acrescentar cabeçalhos de outras funções.
oper.c	Código fonte das funções especificadas em oper.h .

Observações

- O limite de submissões no SuSy é 15.
- Arquivos de teste serão fornecidos para o estudante validar seu programa. A avaliação no sistema será feita com esses e outros testes privados.
- Dúvidas podem ser esclarecidas nas aulas de laboratório ou no grupo do curso indicado no cabeçalho deste documento.

Critério de avaliação

Este laboratório **não vale nota**. Ainda assim recomenda-se fortemente que seja implementado e submetido ao SuSy para relembrar o que foi aprendido em MC102 e para treinar o uso do ambiente de submissão SuSy.

Considerações finais

- Embora existam várias maneiras de resolver os problemas indicados nos laboratórios, o estudante deve optar pela maneira que melhor aplique os conceitos trabalhados em sala de aula.
- Casos de plágio acarretam **média final zero** para todos os envolvidos, sem exceção. O SuSy pode detectar casos de plágio no código, portanto evite compartilhar seu código com outros estudantes, mesmo que seja apenas um trecho.