



EXERCÍCIO 12 – GERENCIADOR ESTRATÉGICO (PESO 2)

- **Prazo para entrega:** 21/05/2017 – 23:55:00
- **Atenção:**
 1. **Arquivo:** o nome do arquivo referente ao código-fonte do exercício deverá respeitar o seguinte padrão: <número do RA>_EX<número do exercício>.c. Exemplo: 123456_EX12.c;
 2. **E/S:** tanto a entrada quanto a saída de dados devem ser “secas”, ou seja, não devem apresentar frases explicativas. Siga o modelo fornecido e apenas complete as partes informadas.
 3. **Identificadores de variáveis:** escolha nomes apropriados;
 4. **Documentação:** inclua comentários e indentação no programa.

- **Descrição**

Em um futuro não muito distante, a revolução das máquinas começou. A maioria da humanidade (em especial os programadores, as maiores ameaças para as máquinas) foi varrida da Terra rapidamente. Os poucos humanos que sobraram vivem em uma batalha constante para resistir aos ataques contínuos dos Exterminadores, ciborgues criados especialmente para acabar com o restante dos sobreviventes.

Do lado dos humanos, a Resistência luta pela sobrevivência enquanto procura uma forma de dar fim aos monstros de metal. Os estrategistas são responsáveis por analisar dados coletados de vários locais, mapeando-os em matrizes que contém em suas posições números que representam o nível de segurança em determinado lugar.

Devido à baixa quantidade de humanos, a maioria das posições dessas matrizes acaba contendo o valor 0, que representa a ausência de humanos e exterminadores. Posições com valores positivos indicam o nível de segurança para os humanos naquele lugar, em contrapartida os negativos representam um nível de perigo para as pessoas.

Com estas matrizes os estrategistas gerenciam de melhor forma os recursos disponíveis (soldados, equipamentos e suprimentos) para garantir a segurança dos sobreviventes desse mundo em ruínas. No entanto, eles acabam perdendo muito tempo com operações simples nas enormes matrizes, devido a falta de um modo automático para realizá-las.

Esta mensagem está sendo enviada do futuro. Você, programador do passado, é a nossa última esperança. Precisamos que desenvolva um programa que realize as seguintes operações sobre as matrizes estratégicas a fim de agilizar o trabalho de nossos estrategistas:

1. **Inserção de matriz:** O usuário informa o índice $1 \leq m \leq 4$ da matriz e dois inteiros $1 \leq l, c \leq 10^4$, representando o número de linhas e colunas da matriz, respectivamente. Seguem então os elementos não-nulos da matriz, cada um em uma nova linha com três valores: i, j naturais e um valor decimal d , os dois primeiros representando a linha e a coluna da posição, nesta ordem, e o terceiro o valor ali presente. A entrada termina quando $i = j = 0$, não haverá valor decimal nesta linha. O programa deverá armazenar a matriz com seu respectivo índice e posições em memória (caso já exista deve substituí-la).
2. **Alteração de posição:** O usuário entra com um índice m de matriz, uma coordenada i, j (linha, coluna) e um valor decimal d . O programa deve armazenar o valor d na posição i, j da matriz m (caso exista).
3. **Soma de linhas:** O usuário entra com três linhas, cada uma com um m e um i , onde m é o índice da matriz e i é o índice da linha na matriz m . As duas primeiras linhas da entrada representam as linhas a serem somadas, e a última representa a linha na qual o resultado deve ser armazenado. O programa deve computar o resultado da soma das linhas e armazenar na linha destino.
4. **Soma de colunas:** Mesmo funcionamento da soma de linhas (opção 03), entretanto as operações são realizadas sobre colunas (i passa a representar o índice de uma coluna nas três linhas da entrada).
5. **Busca por máximo:** O usuário entra com o índice m de uma matriz, um caracter c informando se o índice seguinte é de uma linha ou uma coluna ('L' ou 'C', respectivamente) e um índice i da linha ou coluna desejada da matriz m . Deve ser impresso o valor máximo encontrado na linha ou coluna solicitada.
6. **Busca por mínimo:** Recebe as mesmas entradas que a busca por máximo (opção 05), entretanto o valor encontrado e impresso deve ser o mínimo da linha ou coluna solicitada.
7. **Soma de matrizes:** O usuário informa três índices de matriz m, n e o , nesta ordem. Deve ser calculada a soma de m e n , guardando o resultado em o .
8. **Transposição:** O usuário informa dois índices de matriz m e n , nesta ordem. O programa deve então transpor a matriz m e armazenar o resultado em n .
9. **Exibição de posição:** O usuário entra com um índice m de matriz e uma coordenada i, j . O programa então imprime o valor que está contido naquela posição da matriz m .
10. **Exibição de linha/coluna:** Recebe as mesmas entradas que a busca por máximo (opção 05), e então exibe a linha/coluna selecionada na tela em uma linha, com um espaço após cada elemento exceto o último.
11. **Sair:** Libera a memória e finaliza o programa.

Cada estrategista pode ser responsável por no máximo 4 matrizes, identificadas com índices de 1 a 4. A soma das quantidades de locais habitados por humanos (e consequentemente por exterminadores) de todas as matrizes de um estrategista não é maior que 4000.

Caso seja informada uma posição inexistente na matriz selecionada nas opções 01, 02 ou 09 deve ser exibida a mensagem pré-definida no código base (no caso da opção 01 a mensagem

desse ser exibida, mas a entrada deve seguir, e tanto na opção 01 quanto na 02 o valor decimal deve ser lido e ignorado).

Se uma linha ou coluna for informada com índice inexistente nas opções 03, 04, 05, 06 ou 10 deve ser impressa a mensagem pré-definida (linhas/colunas de destino também devem existir previamente, serão substituídas). Todos os dados da entrada da opção ainda devem ser consumidos na geração de uma mensagem, e deve ser exibida no máximo uma mensagem por opção.

Quanto às opções que armazenam resultados em matriz, não é importante que a matriz destino exista, pois deve ser criada uma nova matriz para aquela posição (podendo ser uma matriz operando, que será substituída ao final da operação), entretanto matrizes que sofrem tentativa de acesso à posições ou que participam de operações como operandos devem existir (mensagem de não existência de matriz tem precedência sobre todas as demais).

Caso as posições e matrizes sejam válidas mas haja incompatibilidade entre os operandos (tamanhos de linhas diferentes na soma de linhas, ou dimensões de matrizes diferentes na soma de matrizes, entre outros) exiba a mensagem definida no código base. Garantidamente não serão realizadas somas de matrizes com dimensões maiores que $10^3 \times 10^3$.

Os limites de número de linhas e colunas na opção 01, caracteres de seleção entre linha e coluna e índices de matriz são sempre informados corretamente, não é necessário verificá-los. Imprima decimais com duas casas de precisão após a vírgula.

Complete o arquivo ex12.c

Exemplo de E/S (os comentários entre parênteses não deverão ser exibidos):

Entrada	Saída
1 (inserir matriz)	
1 5 5 (índice da matriz e dimensões)	
1 1 15.5 (lista de posições não-nulas)	
1 3 8.2	
5 3 7.4	
0 0 (fim da entrada de posições)	
2 (alterar posição)	
1 1 5 (matriz 1, linha 1 e coluna 5)	
3.0 (valor)	
3 (soma de linhas)	
1 1 (matriz e linha do operando 1)	
1 5 (operando 2)	
1 4 (destino)	
5 (busca por máximo)	
1 L 4 (linha 4 da matriz 1)	
	15.60
7 (soma de matrizes)	
1 1 2 (matriz 2 = matriz 1 + matriz 1)	
9 (exibição de posição)	
2 1 3 (matriz 2, linha 1 e coluna 3)	
	16.40
10 (exibição de linha/coluna)	
1 C 3 (coluna 3 da matriz 1)	
	8.20 0.00 0.00 15.60 7.40
11 (sair)	

• **Cuidados:**

1. **Usar TAD:** matrizes esparsas através de listas encadeadas.
2. **Erros de compilação:** nota **zero** no exercício.
3. **Tentativa de fraude:** nota **zero na média** para todos os envolvidos.