

Questão. Uma matriz esparsa é uma matriz onde a quantidade de elementos nulos é superior a de elementos não-nulos. Por exemplo:

$$A = \begin{pmatrix} 50 & 0 & 0 & 0 \\ 10 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 \\ -30 & 0 & -60 & 5 \end{pmatrix}$$

Tal matriz pode ser representada como um vetor de estruturas do tipo **Elemento**:

- linha (caractere sem sinal), coluna (caractere sem sinal) e valor (real)

Por exemplo, a matriz A acima seria representada assim: $A = \{(0, 0, 50), (1, 0, 10), \dots, (3, 3, 5)\}$. Esta representação é usada para economizar memória. Se A fosse armazenada convencionalmente, gastaria 8 bytes (float) x 4 x 4 = 128 bytes. Usando um vetor de **Elemento**, gastaria 8 bytes (**Elemento**) x 6 = 48 bytes. Uma economia de 62,5%.

Faça um programa em C que contenha as seguintes funções:

- **scanMatEsp(int *tam)**: lê três números sucessivamente do teclado com a linha, coluna e valor do elemento da matriz esparsa e armazena em vetor dinâmico de estruturas **Elemento** chamado **NaoNulo**. Esta função retorna este vetor preenchido, com tamanho retornado a partir do ponteiro **tam**. Como a linha e a coluna são representados por caracteres sem sinal, use uma função **validaRange** que garante (validação) que a linha e a coluna estão entre 0 e 255;
- **compensaVet(M, tam)**: esta função retorna **true** (1) se for mais econômico representar a matriz como vetor de **Elemento** e **false** (0) caso contrário;
- **vet2Mat(M, tam, lin, col)**: esta função transforma vetor de **Elemento** em matriz como ponteiro duplo, onde **lin** e **col** são ponteiros para caractere sem sinal;
- **printMatEsp(M, tam)**: apresenta a matriz **M** (como vetor de **Elemento**), devidamente formatada (quadrada/retangular), com elementos nulos nos locais certos, onde o vetor de estruturas tem tamanho **tam**;
- **printMatPontDup(M, lin, col)**: apresenta a matriz **M**, sendo **M** ponteiro duplo;
- **salvaMatEsp(M, tam)**: salva vetor de estruturas no arquivo binário **matEsp.bin**, onde o primeiro byte deve ser 1 para informar que se trata de vetor;
- **salvaMatPontDuplo(M, lin, col)**: salva matriz no arquivo binário **matEsp.bin**, onde o primeiro byte deve ser 0 para informar que se trata de matriz.

Seu programa principal deve:

1. Chama a função **scanMatEsp(&tam)**;
2. Chama a função **compensaVet(M, tam)** em um IF;
 - a. Se o resultado for **true**, chama a função **printMatEsp(M, tam)** e depois a função **salvaMatEsp(M, tam)**;
 - b. Se o resultado for **false**, chama a função **vet2Mat(M, tam, lin, col)**, depois a função **printMatPontDup(M, lin, col)** e finalmente a função **salvaMatPontDuplo(M, lin, col)**.

Boa sorte!!!