Fundamentos da Programação

Prova 1 - parte II

4. (2,0 pontos) **Triangulação embaralhada**

Imagine que alguém pegou uma matriz triangular (que contém apenas `0`'s abaixo da diagonal), e embaralhou as suas linhas e colunas.

Por exemplo,

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 5 & 6 & 7 \\ 0 & 0 & 8 & 9 \\ 0 & 0 & 0 & 10 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} 8 & 9 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 3 & 4 & 1 & 2 \\ 6 & 7 & 0 & 5 \end{bmatrix}$$

Quer dizer, a transformação acima foi realizada por meio das seguintes operações

- `troca das linhas 1 e 3` — ($\ell_1 \leftrightarrow \ell_3$)
- `troca das linhas 2 e 4` — ($\ell_2 \leftrightarrow \ell_4$)
- `troca das colunas 1 e 3` — ($c_1 \leftrightarrow c_3$)
- `troca das colunas 2 e 4` — ($c_2 \leftrightarrow c_4$)

A sua tarefa é escrever um programa que desembaralha a matriz, restaurando o seu formato triangular original

$$\begin{bmatrix} 8 & 9 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 3 & 4 & 1 & 2 \\ 6 & 7 & 0 & 5 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 5 & 6 & 7 \\ 0 & 0 & 8 & 9 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

Quer dizer, o seu programa deve realizar essa tarefa realizando apenas as operações

- `troca de linhas`
- `troca de colunas`

Além disso, o seu programa vai ter que descobrir qual é a sequência de operações que deve ser aplicada, pois a maneira como a matriz foi embaralhada é desconhecida.

Finalmente, o seu programa deve funcionar com matrizes com dimensão arbitrária $n \times n$.

Dica: Faça primeiro uma versão do programa que trabalha com matrizes 4×4.

E depois você modifica o programa para que ele funcione com matrizes de qualquer tamanho.

5. (3,0 pontos) **Tabela de conhecidos**

Imagine que L é uma lista de registros com o seguinte formato

```
struct pessoa
 {
    char  nome[50];
    int   codigo;
 }
```

Por exemplo,

L

maria	pedro	ana	ivo	maria	juca	bia	pedro
27	17	14	31	11	44	28	51

Note que duas pessoas podem ter o mesmo nome, mas os códigos de pessoas diferentes são sempre diferentes.

Imagine também que T é uma lista de registros com o seguinte formato

```
struct amizade
 {
    int   cod1;
    int   cod2;
 }
```

Quer dizer, a ideia é que os registros da lista `T` indicam pares de pessoas que se conhecem e são amigas uma da outra.

Por exemplo,

T

17	27	31	28	44	14	28	51	44
27	11	14	31	51	44	27	17	28

- o primeiro registro indica que `maria` (27) e `pedro` (17) são amigos
- o segundo registro indica que `maria` (27) e `maria` (11) também são amigas
- os próximos dois registros indicam que `ivo` (31) é um amigo comum de `ana` (14) e `bia` (28)

e por aí vai ...

a) Faça um programa que imprime o nome da pessoa com código `c`, e imprime também o nome de todos os seus amigos.

No exemplo acima, se `c` = 27 o seu programa deveria imprimir algo como

```
maria:  pedro,  maria,  bia
```

b) Faça um programa que, dados dois códigos `c1`, `c2`, imprime os nomes das duas pessoas, e imprime os nomes de todos os seus amigos comuns.

No exemplo acima, se `c1` = 14 e $c2 = 28$ o seu programa deveria imprimir algo como

```
ana e bia:  ivo,  juca
```